

Phonetic languages approach to discovering preferred line options (PhLADiPreLiO) using Haskell

Oleksandr Zhabenko
ORCID: 0000-0002-4330-2506

August 9, 2022

Author and software developer: Oleksandr Zhabenko
License: MIT

Copyright (c) 2020-2022 Oleksandr Zhabenko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

The paper proposes an original approach to the task of constructing texts with given phonetic properties, which allows you to work efficiently and quickly on the problem, offering interesting solutions in both content and engineering. Elements of the theory are constructed, which is illustrated by the use of Haskell programs to significantly speed up the necessary calculations.

Keywords

Phonetic languages; prosodic languages; phonetics; prosody; syllables; rhythm; metre; verse; text; coherent states; syllable-as-a-whole (SaaW); phonetic-phenomena-as-a-whole (PhoPaaW); Haskell; phonetic languages approach to discovering preferred line options – PhLADiPreLiO – phladiprelío – the pronunciation is phlah - deeh - preh - 'leeh -oh, the emphasis (dynamic stress) on the second 'i').

Introduction

There are different languages. They have different structure and rules. It is possible to create and use (based on one of existing widely used and well-spoken languages, in particular Ukrainian in this work) a 'phonetic' language that is better suited for poetry and music. It is even possible to create different versions of phonetic language. This paper proposes to create several different phonetic languages based on Ukrainian.

What does this mean?

If someone builds a phrase in a language that violates the rules of grammar or semantics, then this error is visible to a skilled speaker at once, it is identified as such almost instantly. Instead, if the sound of a phrase has some phonetic features, not counting accents, for example, the complexity of pronunciation or vice versa lightness, smoothness or abruptness, etc., then it is possible to identify it as an error or something significant not immediately or with special attention. One can imagine this as giving preference to the language semantics (meaning) and grammar, but less weight to phonetics. Phonetic language is that one built specifically to enhance the meaning and importance of the phonetic component itself.

Phonetic or prosodic language?

An interesting question is whether to call the approach "phonetic" or "prosodic" languages [20]. But I must say that we study the actual phonetic features, what is associated with the sound of speech. Among them is that which concerns certain phonetic phenomena in the general case, in particular phonemes or even palatalization. These questions are generally not the subject of the study of prosody as a science, as a certain component of phonetics, but are the subject of a broader study of phonetics. Moreover, there are no restrictions and bindings of the proposed approach to the actual syllables, which is more typical for the subject of the study of prosody. Generalizations in the package `phonetic-languages-simplified-generalized-examples-array` can be made for more general

cases.

However, at this stage of development, the vast majority of information here relates to or is directly related to syllables and prosody. Therefore, I leave the name "phonetic languages", given that prosody is a more specific branch of phonetics.

Ethical component

The proposed approach is similar to the approach of music theory. Thus, in music, among all sounds, musical ones stand out, later consonances and dissonances are studied, later notes, intervals, chords, melodies, composition, etc. There are recommendations, but they do not bind the creators, but help. Similarly, the proposed approach is designed to provide such assistance. Its strangeness at first glance cannot be a reason to deny it.

For Christians, to whom the author himself belongs, the words of Moses are important: "And Moses went out, and spake unto the people the word of the LORD, and gathered together seventy men of the elders of the people, and set them near the tabernacle. And the LORD came down in the cloud, and spake unto him, Two of the men remained in the state, one named Eldad and the other named Modad, but the Spirit also rested on them. And the young man ran, and told Moses, and said, Eldad and Modad are prophesying in the camp: And Joshua the son of Nun, the servant of Moses, one of his chosen ones, answered and said, My lord Moses. But Moses said unto him, Hast thou not jealous of me? O that all the people of the LORD should be prophets, if the LORD would send His Spirit upon them". (Numbers 11:24-29).

It is good that everybody can well write and speak.

First idea

Imagine that you can understand the information in the text regardless of the order of the words and preserve only the most necessary grammar (for example, the rule does not separate the preposition and the next word is preserved). Understand just like reading a text (after some learning and training, perhaps), in which only the first and last letters are preserved in words in their positions, and all the others are mutually mixed with each other. So imagine that you can understand (and express your thoughts, feelings, motives, etc.) the message of the text without adherence to strict word order. In this case, you can organize the words (keeping the most necessary grammar to reduce or eliminate possible ambiguity due to grammar, or rather a decrease in its volume), placing them so that they

provide a more interesting phonetic sounding. You can try to create poetic (or at least a little more rhythmic and expressive) text or music.

It can also be an inspiring developmental exercise in itself. But how could you quickly find out which combinations are more or less fit? Also, can the complexity of the algorithms be reduced?

These are just some of the interesting questions. This work does not currently provide a complete answer to them, but is experimental one and a research, and any result of it is valuable.

Ukrainian is a language without strict requirements for word order in a sentence (although there are some established preferred options) and has a pleasant sounding. So, it can be a good example and instance. In addition, it is a native language for the author of the programs. Even if you don't want to create and use 'phonetic' languages where phonetics is more important than grammar, then you can assess the phonetic potential of the words used in the text to produce specially sounded texts. It can also be valuable and helpful in writing poetry and possible other related fields [51].

Sound Representations Durations as the Basis of the Approach

There is the fact as the basis of the approach that the language sounds have different durations, which depends on the many factors e. g. mean of the phoneme producing (the different one for every one of them), other factors that can be more or less controlled but usually the full control is not required and is not achieved. This leads to the fact that chaining of the phonemes and phonetic phenomena sequentially among which there is also their syllables grouping introduces some rhythmic painting (picture, scheme). A human can (that is also trainable and can be developed) recognize the traits of the picture, compare them one with another, come to some phonetic-rhythmic generalizations and conclusions.

The question of determining the duration of speech sounds is not easy, but the exact result as already mentioned is not required. In this implementation of the approach of phonetic languages, certain statistical characteristics of sounds are used, in particular, possible durations are determined. If we compare the method of determining durations, which is proposed and used in the program of the r-glpk-phonetic-languages-ukrainian-durations package, the analogy will be the packaging of bulky objects. For the observer, the packaging will be an imaginary model of the process of obtaining sound durations. The pldUkr program (its generalization pldPL from the phonetic-languages-phonetics-basics package also follows this path, but it does not have normalization, because for different languages there may not be such a phenomenon as palatalization) uses linear programming to find the minimum convex hull (not in a strict mathematical sense), which can 'contain' the sounds of speech. This convex hull has an analogy of packaging, while the sounds

of speech have an analogy of objects of variable volume inside the package. The same sound can be used in different situations, in different words with different durations, but the program tries to choose such durations that would 'cover' (similar to the envelope curve 'covers' a family of curves) all these variations for all sounds, with an additional normalization of the duration of the phonetic phenomenon of palatalization (softening) of the consonant, which is least controlled by man, and therefore it is expected that this duration is the most resistant to possible random or systematic fluctuations. For the Ukrainian language the possible duration which does not change strongly sounding is defined experimentally with use of the computer program mm1.

Finally, normalization is not mandatory, it is important that all durations are proportional to each other, i.e. it is not the durations themselves (which are numerically expressed as real positive numbers) that are important, but their mutual ratios (it is allowed to multiply these durations simultaneously by one and the same positive number that does not affect the results of the approach).

Polyrhythm as a Multi-Ordered Sequence Pattern

Let us have some sequence organized in the following way. Let us implement (generally speaking a conditional one) division of the sequence into compact single-connected subgroups with the same number of elements each in the subgroup, which actually means that we split the sequence into a sequence of subsequences with the same number of elements in each. Consider the internal ordering of each subsequence from the perspective of the placement of the values of its elements and repeatability of the some patterns of the placement of the elements. We assume that the elements can be compared in relation of order, that is, they are the elements of the data type that has an implemented instance of the class Ord.

Considering that the elements of the subsequences may be pairwise different (or in some cases equal), we will compare the positions on which the subgroups of elements that have a higher degree of relatedness ("closeness", "similarity") in value and order are located. Denote such subgroups by indices that have in the module code mostly a letter designation.

Then each subsequence will consist of the same number of elements of one nature (in particular, numbers of the type Double), in each subsequence there will be selected several subgroups of "similar" elements in value (and order, if the subsequences are sorted by the value), each of which will have its own index as a symbol (most often in the code – the characters). Subgroups must have (actually approximately) the same number of elements (in the code it is not strictly used for simplification of the former one, but it is so in the vast majority of cases because of the excessive "accuracy" of numbers of type Double that are used). Consider the question of positions in the subsequences of the corresponding subgroups in case of they have been belonging to different subsequences.

To assess this, we introduce certain numerical functions that have regular behavior and allow us to determine whether the sub-

sequences actually have elements that belong to the relevant corresponding subgroups in the same places, or on different ones. It can be shown that the situation "on different ones" corresponds to the presence of several rhythmic patterns - for each subgroup will be their own, which do not mutually match, at the same time the ideal situation "completely in the same places" corresponds to the case when these rhythms are consistent with each other, as is the case of coherence in quantum physics, in particular spatial and temporal coherence, which is important in particular for understanding of lasers and masers. Polyrhythms consisting of such rhythms, which cohere with each other, form a more noticeable overall rhythm, as well as the presence of coherence in the radiation leads to a more structured latter one [52].

As an illustration for the ideas of the section the following data.

An example of the rhythmic sequence (almost ideal case).

```
Prelude Rhythmicity.PolyRhythm Numeric> let f x = putStrLn . showFFloat (Just 4) (sin (2*pi*x)) $ ""
      in mapM_ f [0,0.2..4]
0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
0.5878
```

8

```
-0.5878  
-0.9511  
-0.0000
```

```
Prelude Rhythmicity.PolyRhythm Numeric> getPolyChRhData 'a' 5 (PolyCh [True,True,True,False] 5)  
  (PolyRhythm [1,1,1,1,1]) . map (sin . (*pi) . (*2)) $ [0,0.2..4]  
[[RP P c,RP P a,RP P b,RP P e,RP P d],[RP P c,RP P a,RP P b,RP P e,RP P d],[RP P c,RP P a,  
  RP P b,RP P e,RP P d],[RP P c,RP P a,RP P b,RP P e,RP P d]]
```

Here is the example of the sequence with not stable rhythm or with the rhythm that is less evident.

```
Prelude Rhythmicity.PolyRhythm Numeric> let f x = putStrLn . showFFloat (Just 4)  
  (sin (27182.81828459045*pi*x)) $ "" in mapM_ f [0,0.01..0.24]  
0.0000  
-0.5139  
-0.8817  
-0.9988  
-0.8319  
-0.4284  
0.0969  
0.5947  
0.9233  
0.9894  
0.7742  
0.3388  
-0.1930  
-0.6698  
-0.9562  
-0.9707  
-0.7092
```



```

-0.2460
0.2872
0.7386
0.9801
0.9428
0.6375
0.1509
-0.3787

```

```

Prelude Rhythmicity.PolyRhythm Numeric> getPolyChRhData 'a' 5 (PolyCh [True,True,True,False] 5)
(PolyRhythm [1,1,1,1,1]) . map (sin . (*27182.81828459045) . (*pi)) $ [0,0.01..0.24]
[[RP P a,RP P b,RP P e,RP P d,RP P c],[RP P d,RP P e,RP P c,RP P b,RP P a],[RP P a,RP P b,
RP P c,RP P e,RP P d],[RP P d,RP P e,RP P c,RP P b,RP P a],[RP P a,RP P b,RP P c,RP P e,RP P d]]

```

Coherent States of Polyrhythmicity as One of the Essential Sources of Rhythmicity

The described pattern of rhythmicity is one of the significant possible options for the formation of rhythmicity in particular in lyrics or music, but not the only one. It should be noted that the described mechanism of rhythm formation, as is noticed in the statistical experiments with texts using this code (the code of the library and its dependent packages on the Hackage site) may not be the only possible option, but in many cases it is crucial and influences the course of the rhythmization process (formation, change or disappearance of the rhythm). It is also known that the presence of the statistical relationship does not mean the existence of deeper connections between phenomena, in particular – the causality. "Correlation does not mean causality." A deeper connection implies the presence of other than the statistical ones to confirm it.

Rap Music Consequences

The code of the library allows in practice to obtain rhythmic patterns that are often close to the lyrics in rap style. Therefore, this can be attributed to one of the direct applications of the library.[52]

A Child Learns to Read, or Somebody New to the Language

When a child just begins to read words in the language (or, there can be just somebody new to the language) he or she starts with phonemes pronunciation for every meaningful written (and, hence, read) symbol. Afterwards, after some practice, he / she starts to read smoothly. Nevertheless, if the text is actually a poetic piece, e. g. some poem, it is OFTEN (may be, usually, or sometimes, or occasionally, etc.) just evident that the text being read in such a manner has some rhythmicity properties, despite the fact that the phonemes are read and pronounced in a manner of irregular and to some extent irrelevant to the normal speech mode lengths (durations). We can distinguish (often) the poetic text from the non-poetic one just by some arrangement of the elements.

The same situation occurs when a person with an accent (probably, strong, or rather uncommon) reads a poetic text. Or in other situations. The library design works just as in these situations. It assumes predefined durations, but having several reasonable (sensible) ones we can evaluate (approximately, of course) the rhythmicity properties and some other ones, just as the algorithms provided here.

This, to the mind of the author, is a ground for using the library and its functionality in such cases.[53]

Increasing and Decreasing Functions

Since the 0.5.3.0 version of the phonetic-languages-rhythmicity package the increasing and decreasing functions for the polyrhythmicity evaluation have become more similar to be more likewise the inverse one to another. This leads to that fact that these functions now are expected to be smoother for usage for the beginning of the line, its middle and its ending.

Note: since the 0.6.0.0 version of the phonetic-languages-rhythmicity package the values of the properties from the series “c”, “s”, “t”, “u”, “v” and the many others (starting from the 0.9.0.0 version) can be negative by sign. This does not influence the logics of the working library functions and programs. Since the 0.8.0.0 version of the package phonetic-languages-simplified-examples-array there were added also new properties that can be negative by sign.

Problem of choosing the best function and related issues

Consider the following question: suppose we have obtained the best version (in our subjective opinion or on the basis of some criteria, it is irrelevant here) of the line in one way or another (here the method does not really matter). Is there a function that makes this particular variant of the string optimal, i. e. for which such a variant of the string gives the maximum of all possible permutations? Yes, there is. This is easy to prove. The proof is reminiscent of the principle of equalizers.

Let $n \in \mathbb{N}$ be the number of syllables in such a line. Arrange the durations of the syllables in ascending order (standard procedure for descriptive statistics). We will find the smallest nonzero difference between adjacent values, divide it by 5. Denote this value by δ . Now consider a number of syllable durations for our best string. Number each syllable from the beginning, counting from 1. Denote by $Y = \{y_i, \quad i \in N, \quad i = 1, 2, \dots, n\}$ the set of all values of durations in the order of sequencing in the best line. Denote by $X = \{0 = x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{n+1}, \quad i \in N, \quad i = 1, 2, \dots, n\}$ the set of coordinates of the points of the ends of time conventional intervals, into which our best line divides the time line (the left edge is 0, because the countdown starts with 0). Denote by $M = \{z_i = \frac{x_i + x_{i+1}}{2}, \quad i \in N, \quad i = 1, 2, \dots, n - 1\}$ a set of midpoints of the segments into which the time line is divided by the conditional intervals ends. Denote by $L_1[a, b]$ the class of Lebesgue-integrable functions on the interval $[a, b]$. Let's mark $I(y_i, z_i, \delta)[z_i - \delta, z_i + \delta]$ class of all bounded functions with $L_1[z_i - \delta, z_i + \delta]$, the maximum and minimum values of each of which lie on the segment $[y_i - \delta, y_i + \delta]$. We denote each function of class I by g .

Consider the class of functions F (a kind of finite analogues of the known delta functions of Dirac), defined as follows:

$$f(x, i) = \begin{cases} g \in I(y_i, z_i, \delta)[z_i - \delta, z_i + \delta], & \text{if } y_i \text{ is a unique value in the set } Y, x \in [z_i - \delta, z_i + \delta] \\ y_i, & \text{if } y_i \text{ has equal value with some other number from the set } Y, x \in [z_i - \delta, z_i + \delta] \\ 0, & \text{otherwise} \end{cases}$$

It is easy to see that

$$\sum_{i=1}^n \int_{-\infty}^{\infty} f(x, i) \, dx,$$

where integration is carried out according to Lebesgue, is the desired function (because only the syllables of the best string in their places, taken into account with their indices, give a positive contribution to its value, and for all other variants of the function at least some of the syllables give 0 contribution), and it is not unique due to the fact that in the first line of definition $f(x, i)$ the function can

have an arbitrary value from a closed non-empty interval. Therefore, there is at least one class of functions that is described by such a formula, for each of which this particular variant of the string will be optimal.

Let's ask the following question: if we consider not a line, but their combination, for example, a poem. Does a function that makes each line optimal (i. e. which will describe the whole work, each line in it) exist for the whole work (for this whole set of lines)?

In this case, the previous method of constructing the function does not give the desired result, because for two lines it may be that what is best for one of them is not the best for the other. In the case of increasing the number of lines, this general suboptimality only intensifies. However, the existence of such a function for different particular cases is a fundamentally possible situation, however, the probability of its existence should decrease both with increasing number of rows and with the appearance of different features of rows, which increase the differences between them. In general, the search for just one such function may be virtually impractical.

Nevertheless, if we consider the whole work as one line, considering it optimal, then the method just described to construct the corresponding function (class of functions) again gives the result. Yes, the number of syllables in it (in the generalized "line"-work) increases, but the procedure itself gives similar results (if we neglect the possible identical durations and repetitions of syllables in a larger text, which lead to the fact that some words can be rearranged and that makes the text only approximately optimal). It is possible to suggest further improvement of this procedure (for example, introduction of the factors which depend on values of the next durations of syllables) that allows to reduce suboptimality of the text.

But still, if you go from one text to another, the resulting function is unlikely to be optimal.

We conclude that for the whole set of expediently organized texts the existence of a universal function is seen as a certain hypothesis (with almost certainly proposes a negative answer).

Connection with fractals

We will pay attention to this fundamental "break", the fundamental dissimilarity of the texts.

But first note the following. The function considered above makes one line variant optimal at the same time you can consider not one option, but several, from which to choose the best. Then the task is reduced to search optimal group of options with the minimum possible number of components.

If we consider again the boundary case of a group with the number of elements equal to the factorial of the number of words in a line, it is easy to see for combinatorial reasons that such a group contains (consists of, is) all variants of a string of these words. But for 5 words such a group contains $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ options for 5 words, a total of 600 words to read which usually takes a few

minutes.

If you reduce the number of words in the optimal group, then among them may not be the right option, but the reading time decreases accordingly. So, if we have some way of at least approximate ordering, it will be expedient to find the optimal ratio "group size - the degree of accuracy of approximate ordering".

This leads to the idea of using different functions that are easy to calculate, have a certain characteristic behavior and allow you to roughly organize the sets of all permutations according to the search task, and the search is no longer a single option, but the optimal group of options.

At the same time, the search stages are fundamental, when the program does not give the desired result completely. Then you can change something in the data (without changing the words themselves), or change the words to change the options that the program works with, because it changes the structure of the set of all options and allows you to get other still somewhat "optimal" options from the author's point of view and in terms of the program. This gave impetus to the so-called recursive mode of operation, when the change of data is the merging of words (and hence the exclusion from consideration of options where they are not in a row), as well as the mode of several options.

An interesting observation is also that in recursive mode you can get lines in which more noticeable are the "breaks" of rhythm, which can often be included in the rhythm through the introduction of additional pauses. Poems with pauses (for example, so-called caesuras) are also known in theory, for which this may be useful.

Also the ability to form lines from lines, and each line from certain components, resembles fractals. However, the relationship with fractals requires more detailed and in-depth study.

An interested reader can turn to the literature.[7, 33, 31, 27, 48, 19, 8, 37, 49, 18, 11, 3, 30, 45, 39, 2, 6, 29, 42, 24, 4, 71, 76]

Analogy with the extremity principle

In mechanics and optics, the principle of extremity of action is known (a physical quantity of action is introduced, which for real processes in these areas takes among all possible values along the trajectories of the minimum (most often), or maximum value, or the constant one with respect to the trajectory, ie in one word the extreme value). In thermodynamics there is a law of increasing entropy of closed systems. In the case of mechanics and optics, the search for real trajectories is reduced to finding those of the possible trajectories for which the value of the action functional is extreme, which allows you to use the apparatus of higher mathematics to find these extreme values.[47]

It was suggested that the analogy with this principle, that is, the comparative mode of operation in combination with the mode of several properties, can give the best result. However, a wide range of good rhyming lines (and not only them) do not necessarily follow this rhyming pattern. Therefore, in general, the principle serves as a kind of search intuition, but not as a regularity.

Ability to use your own durations of representations of sounds or phonetic phenomena

The programs offer four different sets of phonetic representations by default but starting with version 0.13.0.0 it is possible to set your own durations. To do this, specify them as numbers of type 'Double' in the file in the order defined as follows:

| | | |
|----------|----------------------|----|
| UZ 'A' D | дз (plain) | 8 |
| UZ 'A' K | дз (palatalized) | 9 |
| UZ 'B' D | ж (plain) | 10 |
| UZ 'B' K | ж (semi-palatalized) | 11 |
| UZ 'C' S | й | 27 |
| UZ 'D' N | сь | 54 |
| UZ 'E' L | ч (plain) | 39 |
| UZ 'E' M | ч (semi-palatalized) | 40 |
| UZ 'F' L | ш (plain) | 41 |
| UZ 'F' M | ш (semi-palatalized) | 42 |
| G | | 55 |
| H | ю | 56 |
| I | я | 57 |
| J | є | 58 |
| K | ї | 59 |
| L | ' | 60 |
| M | ' | 61 |
| N | HT | 62 |
| O | CT | 63 |
| P | ТЬ | 64 |

| | | | |
|----|-------|----------------------|----|
| | Q | ДЗЬ | 12 |
| | R | зь | 13 |
| | S | нь | 65 |
| | T | дь | 14 |
| UZ | 'a' W | a | 1 |
| UZ | 'b' D | б (plain) | 15 |
| UZ | 'b' K | б (semi-palatalized) | 16 |
| UZ | 'c' D | ц (plain) | 38 |
| UZ | 'd' D | д (plain) | 17 |
| UZ | 'd' K | д (palatalized) | 18 |
| UZ | 'e' W | e | 2 |
| UZ | 'f' L | ф (plain) | 43 |
| UZ | 'f' M | ф (semi-palatalized) | 44 |
| UZ | 'g' D | г (plain) | 19 |
| UZ | 'g' K | г (semi-palatalized) | 20 |
| UZ | 'h' D | г (plain) | 21 |
| UZ | 'h' K | г (semi-palatalized) | 22 |
| UZ | 'i' W | i | 6 |
| UZ | 'j' D | дж (plain) | 23 |
| UZ | 'j' K | дж (palatalized) | 24 |
| UZ | 'k' L | к (plain) | 45 |
| UZ | 'k' M | к (semi-palatalized) | 46 |
| UZ | 'l' S | л (plain) | 28 |
| UZ | 'l' O | л (palatalized) | 29 |
| UZ | 'm' S | м (plain) | 30 |
| UZ | 'm' O | м (semi-palatalized) | 31 |
| UZ | 'n' S | н (plain) | 32 |
| UZ | 'n' O | н (palatalized) | 33 |
| UZ | 'o' W | o | 3 |

| | | | |
|----------|----|--------------------|----|
| UZ 'p' L | п | (plain) | 47 |
| UZ 'p' M | п | (semi-palatalized) | 48 |
| UZ 'q' E | ь | | 7 |
| UZ 'r' S | р | (plain) | 34 |
| UZ 'r' O | р | (palatalized) | 35 |
| UZ 's' L | с | (plain) | 49 |
| UZ 't' L | т | (plain) | 50 |
| UZ 't' M | т | (palatalized) | 51 |
| UZ 'u' W | у | | 4 |
| UZ 'v' S | в | (plain) | 36 |
| UZ 'v' O | в | (semi-palatalized) | 37 |
| UZ 'w' N | ць | | 66 |
| UZ 'x' L | х | (plain) | 52 |
| UZ 'x' M | х | (semi-palatalized) | 53 |
| UZ 'y' W | и | | 5 |
| UZ 'z' D | з | (plain) | 25 |
| UZ 'z' K | з | (palatalized) | 26 |

where the specified values in the list refer to the phonetic representations of the data type UZPP2 (from the module Languages.Phonetic.Ukrainian.Syllable.ArrInt8). The last column is 8-bit integers (GHC.Int.Int8), which represent these sounds in the new modules.

If you want to specify several such sets (up to 9 inclusive), you can specify '*' or several such characters from a new line, and then from the next line there will be a new set of values.

Each set should be in the following order: [1,2,3,4,5,6,7,8,9,10,11,15,16,17,18,19,20,21,22,23,24,25,26, 27,28,29,30,31, 32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,66,101]

where the number corresponds to the last column in the above diagram. 101 (prior to the version 0.20.0.0 here there was -1 instead, and it was at the beginning of the list, not at the end) corresponds to a pause between words (does not affect the search results of the line). Every new value must be written in the file from the new line.

Then when executing the program somewhere among the command line arguments (it does not matter where exactly) specify «+d» <path to the file with the specified data>. Programs will read these values and convert them to the appropriate values. As properties

it is necessary to use then those which begin with the letter «H», and further the corresponding designation of property merged with it. For example, «Hw04», the last digit in the record in this case will mean the ordinal number of the set of values, starting from 1 (maximum 9).

Along with the custom values, you can use the provided by the library ones, as usual, in the mode of several properties.

Minimum grammar for possible preservation of meaning and intelligibility

Programs use permutations of words that neglect any (or at least part of) grammatical connections, word order, and so on. This can lead (in addition to the need to think) to situations where grammatically related language constructions are broken, their parts are transferred to other places, forming new connections and changing the meaning of the text.

To reduce this, to eliminate some of these effects, programs use concatenation of words that have a close grammatical connection, so as not to break them in the analysis. This allows you to maintain greater semantic ease and recognizability of the text, as well as a side effect to increase the overall length of the line, which can be analyzed. In the Ukrainian language, grammatically related auxiliary or dependent words precede the independent or main one, so the concatenation of these auxiliary or dependent words to the next one is used. The completeness of the definition of such cases is not exhaustive, but the most frequent cases are considered.

To reduce this, to eliminate some of these effects, programs use concatenation of words that have a close grammatical connection, so as not to break them in the analysis. For the general case, it should be borne in mind that auxiliary or dependent words can go after the independent or the main, so this should be considered separately (and attach such words to the previous, not to the next). Currently, the generalized version of `phonetic-languages-simplified-generalized-examples-array` supports both.

Reduced set of permutations as a variant of the universal one

By default, the program analyzes the universal set of permutations of all words and their concatenations, while the number of analyzed options increases as a factorial of the number of such words or combinations. The text, organized more or less coherently in relation to one or another property, can be radically different from the original, which complicates understanding and has the effect of delaying calculations.

To quickly test the possible improvement of the text using the approach introduced sets of permutations of only one word relative to the text, as well as two words or their combinations as universal sets (in addition to the full one). When running programs, they are specified by the command line parameter «+p» somewhere among the arguments (the position does not matter). In this case, a reduced set of permutations is used. The number of words and their combinations that can be considered by the program as one line for analysis increases in this case to 10. To use the minimum possible set of permutations, you must specify as the next command line argument «1» (only one word), for pairwise permutations – «2» (pairwise permutations). Then among the arguments of the command line will be the expression «+p 1» in the first case, «+p 2» – in the second, and for the full set of permutations, you can still not specify this group (the full set is thus used for default).

In this case, the analysis is much faster (because the number of cases is significantly reduced), and the text changes less, which allows you to keep it more recognizable. Recursive sequential application of this case is possible, but it should be borne in mind that in the case of pairwise permutations this may not be the best option in terms of approach, and it may take even longer than analyzing the whole full set of permutations (because some options will be analyzed for several times, which does not happen in the first case). In the case of permutations of only 1 word relative to the text, it seems that if we recursively sequentially apply the search of the maximum element twice in a row for the same properties and we get the same option, then the chances that it will be maximum for the full set of permutations increase compared to pairwise permutations, but to estimate the increase (if this hypothesis is really true) is difficult in the general case. However, in practice, the search for such a 'minimally improved' option is promising, as it may well retain meaning and in advance – may make rhythmic structure better.

The following should be taken into account: when searching for the maximum element by the value of the property (i.e. without changing the structure) if the analysis of reduced permutation set received a text that coincides with the original, then there is a good chance that this option is optimal in terms of the property in question (although this is not guaranteed). And one more thing: in this case, a local maximum is reached (which may or may not be the global one). If the repeated application leads to the formation of another (already the third) option, then the previous local maximum was not exactly global and the program is moving in its direction.

See also: [73, 55, 72, 70, 75, 54, 69, 74, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]

More about rhythm in music and language

Musical rhythm, meter, pulse and other similar topics are discussed in [12].

Pulse means one of a series of regularly repeating the same stimuli (sounds). The meter is a measure of the number of pulses

between more or less regular accents. Accents are those pulses that for a person seem to stand out among others. Rhythm (musical) there means how one or more unaccented bits (pulses in a metric context) are grouped around one accented one. Accent can occur in different ways and does not coincide with the amplification of sound, and can be caused by a variety of factors in music.

It is noted that there are no strict and simple rules for determining rhythmic groups, but there are certain trends (at least in Western music – *Author's note*) that need to be considered. Sometimes the grouping can be ambiguous (multi-valued), multivariate, and allow different variants to coexist.

Grouping is architectonic, ie has different levels that interconnected. At different levels, grouping is the product of the similarities and differences of sounds, as well as the closeness and separation that are felt by the senses and organized mentally.

I will note here that the programs consider structures with the same number of syllables in one group, but several elements in one group can be selected, almost all. Moreover, the degree of ability to be remarkable is considered, which is considered as a measure of the importance of certain syllables in calculating the values of properties. These groups of syllables can be correlated with the meter, while different numbers of significant syllables correspond to the fact that the meter can exist regardless of rhythm. In the classical approach considered, a group with one accented sound is rhythmic. In the programs, there is no clear relationship between the accentuated composition and the degree of importance. This means that a different approach to this issue is proposed, the results of which can be fruitful.

However, it can often be assumed that a group of significant syllables corresponds to a rhythmic group at one of the architectonic levels, and then the structures with the maximum value of the corresponding properties of this type will correspond to lines with more regular repetition of rhythmic groups in their general features, on this level of composition of the work, and smaller values of the corresponding properties of this type will correspond to unevenly distributed rhythmic groups, or their variations, i.e. more complex (and therefore perhaps less recognizable and tangible, less significant) rhythmic pattern at a higher architectonic level. It can also be tentatively assumed that for the maximum values of the corresponding properties of the strings rhythmic grouping is associated with metric, but if we do not consider the maximum elements, the degree of connection between rhythmic grouping and metric weakens, i.e. such strings tend (in the musical sense) to "free" rhythm of Oriental and folk music and "measured rhythm" of Gregorian chants [12]. In fact, it offers a wide range of opportunities to explore the relationship between metric and rhythmic structure in language.

Differences between rhythmicity in music and language and their cultural conditionality

Instead, [25] shows that instead of grouping based on perceptual principles (louder sound starts the group more often, and lengthened sound or interval ends the group more often), different groupings are possible for music in the English and Japan cultures, as well as patterns in corresponding languages. In the same paper, there are many references to works that have previously considered one or another approach. Also listed differences between English (also Ukrainian is closer to English in this context) and Japanese, in particular the official parts of the language precede the main words in English (and Ukrainian), while follow the main ones in Japanese.

In [21] the differences between languages in syllable durations are mentioned: languages can be divided into those in which the structure of syllables is more or less variable. This affects in particular the unequal duration of sounds in different positions, the phenomena of reduction of vowels in languages where the structure of syllables is more variable. In [5] it is also noted that, for example, in French the phenomenon of lengthening the duration of final syllables is very noticeable, especially at the end of a phrase or sentence. Programs do not take into account such differences in the duration of syllables, but their consideration may be a task for further improvement.

Differences in rhythmicity in music and language are discussed in [28]. It uses an interdisciplinary approach. In particular, the analysis of brain activity in reactions to incongruence (deviations from the structure of rhythm) in language shows that they significantly influence the process of understanding the meaning of language (prolong reaction and comprehension time). Similar results on the influence of rhythm on the ability to understand and speak are given in [32].

In [26] the peculiarities of the African-American rhythmic system in music are considered, in particular, such features as more saturated rhythmic development of smaller parts of works and less variability between smaller parts of one larger work, as well as the idea of expressive rhythm, in which, in contrast to the uniform beat of Western music, an approximate equivalent of a beat and a much shorter "atomic" rhythm are introduced, which, presumably (conditionally), consists of notes of about $1/16 - 1/24$ durations and introduces "expressive details". The phenomenon of superposition is also noted in combining of different cyclic rhythms into one common (well traced, for example, in the Afro-Cuban rumba), which is similar to the idea of coherence of polyrhythms to form a single rhythm. The phenomenon of expressive "atomic" rhythm, its use and characteristics show that it makes sense to enter the duration of phonetic phenomena with high (at first glance excessive) accuracy.

In [50] it is shown that children aged 5-24 months are more responsive to musical rhythms than speech, also that they mostly evoke positive emotions in them, as well as motor activity.

In the work [44] it is considered how rhythms in music can cause emotional states or affect them. It is noted that this is not an automatic process, and it is influenced by individual tastes, familiarity and training.

In the work [9] an attempt is made to investigate the influence of melodic and rhythmic changes on the perception of melody. It was found that melodies unfold more slowly (in perception), if they have more changes in pitch, more incompatible changes in rhythmic structure.

In [14] it is proposed an approach to determining the beginning in a musical sound signal based on the analysis of rhythm as well.

In [46] it is studied how brain processes musical information in case of experienced jazz musicians compared to unqualified ones. It is shown that there are noticeable differences in the ability to predict rhythm. This indicates the possibility of training the sense of rhythm.

In [10], it has been experimentally shown that pre-listening to music with a similar rhythm improves the ability to detect phonemes in speech, and this effect is enhanced by pre-audio-motor preparation for listening to speech sentences. This can be used, for example, in speech therapy. There is also an assumption about the processing of linguistic and musical temporal (rhythmic) structures by the brain with the participation of common (same) resources.

In [15] it is investigated the relationship between the temporal and rhythmic structure of music at the note level, and it is proposed a method that significantly takes into account the rhythm to determine the onshot of the note in the sound signal.

In [41] it is experimentally shown that if it is a meaningful phrase, phonemes in stressed syllables are perceived faster, and if the phrase consists of nonsense words (without lexical meaning), then there is no significant difference in reaction time for stressed and unstressed syllables. It is concluded that stressed syllables can be predicted (variation of apperception), while unstressed – no (this explains the difference in reaction time). This result suggests that the presence of dynamic accents improves the speed of comprehension of the text (for those who are native speakers of a culture with a language that has dynamic verbal accents).

The paper [34] draws some conclusions about the connection between rhythmic structure in language and music, in particular the obvious fact that there is much in common in rhythmic groupings in language and music, but this does not make them identical. For research, it is further proposed to consider the influence of linguistic rhythmic structure on music. It is mentioned that the available data partially confirm that the musical rhythmic structure is influenced by language. Linguistic (linguistic) rhythm is a combination of several factors that affect the temporal organization of speech. First, it is the alternation of words and pauses, secondly, the different durations of syllables, and thirdly, the alternation of stressed and unstressed syllables. These factors can cause languages to be rhythmically similar or different. Musical rhythmic grouping also means grouping into a phrase, whether there is a beat (a periodic one), as well as a metric structure. What is common is the fact of grouping into phrases, and the differences are mainly in periodicity. In particular, the authors emphasize that the early and influential hypothesis about the division of languages into those in which accents are approximately evenly distributed and those in which syllable durations are approximately the same, i.e. the beginnings of syllables are approximately evenly distributed, is not confirmed by available experimental data. This creates real prospects for using

programs for many languages. However, linguists have retained these concepts and continue to describe differences in languages. There are discussions about whether these are really important concepts, or just two ends of a continuum in which all languages have their place. Also important is the fact that in normal conditions there is no periodicity in the speech rhythm, in contrast to the common in music. The authors also express their belief that the influence of language rhythms on music is not universal, but more characteristic of periods and events when composers try to emphasize their national identity and affiliation.

Advice how to use the programs

In correspondence with the previous information, the rhythmicity of the proposed by the programs variants can be in many cases more evident and perceived better if you read the words (their concatenations) at the lines without significant pauses between them (as one single phonetic flow), not trying to strengthen emphases (probably even without well articulated emphases, smoothly, with liaising). This can be even in a different way, but if the obtained variants do not seem to be rhythmic enough then try just this option and since then compare and come to conclusion whether such sounding is just suitable for your situation.

Some additional information

The work [40] proposes a method for classifying music that uses hidden Markov chains, as well as information on rhythmic structure, comparable in accuracy with manual classification.

The work [22] proposes a relatively effective way to determine the rhythmic similarity of music fragments.

In [16] it was studied whether pianists-performers distinguish different features of rhythmic structure, meter and melody by systematic variations of performance. The most characteristic were the allocation of rhythmic grouping.

In [43] the authors try to use wavelet analysis to obtain characteristics of rhythmic structures of musical pieces and speech, highlight their mathematical features, then apply the results to analyze pieces. It is shown that the wavelet characteristics have a perceptual basis. It is proposed to discuss the possibility of applying the obtained results to cascading rhythm generation.

The [38] paper considers a statistically common approach to determining the linguistic rhythmic structure based on vocal (vowels and their sequences) and intervocal (consonants and their sequences) intervals (distances, durations) and their variability. It is shown that this allows better typology of languages in terms of rhythmic structure. It is additionally considered to the already mentioned

languages with approximately the same frequency of accent distribution, languages with approximately the same syllable duration, as well as languages with approximately the same distribution of moras (mora is a syllable with a short vowel or one short vowel). Variants of pairwise variability indices (rPVI, nPVI) and / or standard deviations are used for estimates. Much of the work is devoted to description points of view on the question of whether it is possible to cluster and typify languages in rhythm or that they all form a certain continuum of values, distributed within one large group. Research is ongoing in this area.

The work [17] shows that sonority can be used instead of durations to determine rhythm, which also makes it easier to automate the segmentation process into rhythmic groups. Later works continue to study this question, using, in particular, hidden Markov chains.

In the work [35] it is considered the influence of the peculiarities of the linguistic rhythmic structure on the musical one for the outstanding English and French composers of the past centuries. There is a discussion on this issue.

The work [36] analyzes this issue for German and Austrian composers over a period of about 250 years, pointing out that there are fluctuations that can be explained by historical and cultural rather than linguistic influences.

In [13] the same theme continues in comparison with Italian composers.

Theories of melodic musical emphasis are considered in [23]. Experimental consideration confirms mainly the theory of Joseph Thomassen (1982). In this model (theory) the most emphasized are the turning points in the melody, when the change of pitch occurs in opposite directions, and this is more noticeable for ascending-descending rotation than for descending-ascending. But in different cases there may be a melodic accent of a different kind. It has been shown that melodic emphasis can be a rather weak factor in the rhythmic structure of music.

The work [21] analyzes whether listeners can determine the rhythms of music (the language they know well) in which language the song is. As it turned out, in many cases they can. Listeners can use the similarities and differences in the rhythmic structures of languages to determine which language the song was composed of.

An interesting additional information about the rhythm and related musical topics is in the video:

1. Adam Neely. Solving James Brown's Rhythmic Puzzle. Adam Neely. 2021.

Prerequisites for using the software package

At the moment, the programs work for workstations (desktops e. g.), and there are no mobile versions.

You must have Haskell applications installed and configured:

1. GHC (versions not earlier than 7.10)
2. Cabal

The executables of these programs must be searchable through the PATH workspace variable (this is the default setting).

If possible, use the system package manager (programs) to install also important packages Haskell bytestring, vector, heaps, parallel. If you also plan to use r-glpk-phonetic-languages-ukrainian-durations, then also install and set the programming language (and better development environment) R.

If the required Haskell packages are not installed using the system manager, they will be installed when installing the packages (downloaded and installed automatically, with additional time also spent on their compilation).

Remark on terminology

Earlier versions of the packages used the names 'norms' and 'metrics' for the properties of the texts. Because in the sense typical of mathematics (including functional analysis) all these properties are not actually metrics and norms (e. g. the inequality of the triangle is not fulfilled), then the non-ambiguous 'property' will be used everywhere instead.

'Property' hereafter means the functional representation of the latter.

Installing the package

Open a command prompt or terminal and enter as commands:

```
cabal update
cabal --reinstall --force-reinstalls install phonetic-languages-plus-0.6.0.0
cabal --reinstall --force-reinstalls install phonetic-languages-simplified-examples-array-0.20.0.0
```

It is also recommended to install the following packages:

```
cabal --reinstall --force-reinstalls install r-glpk-phonetic-languages-ukrainian-durations-0.4.2.0
cabal --reinstall --force-reinstalls install mmsyn6ukr-array
```

The latter is optional, but useful for sound and does not take a lot of space.

If there are messages about outdated command variants, enter instead of update – v1-update, install –

```
--enable-split-sections --enable-split-objs --enable-library-stripping \
--enable-executable-stripping v1-install
```

If this does not help, please, wait for the newer package versions that will have the issue being solved, or write to the maintainer to the email address olexandr543@yahoo.com.

In the newest version of cabal it is planned to use v2-* command versions by default, but there are some additional needed workarounds to implement different versions of the packages, libraries and executables, that is why it is recommended to use v1-* variants. For example, if after the installation in such a way (v2*) while loading a module into the interpreter GHCi there are error messages that the module is in the hidden package, then run the interpreter again with a flag -package <name of the module's package>.

The packages base, parallel and heaps can be often found in the OS repositories, the rest of them it is recommended to install from the Hackage server (using the above-mentioned commands).

Changes in version 0.17.0.0

In the new version an attempt is made to optimize the calculation of the conversion of Ukrainian text into data for analysis. Duplicate values for lines 4 and 3 for different property types have also been fixed. If you want to reproduce the previous results for line 4, you now need to use line 3, instead line 4 potentially should give a more recommended result by the author. Version numbering rules have also been changed to allow for more stable releases over time and the ability to reproduce program results for a future versions.

Changes in version 0.18.0.0

In the new version there were fixed issues with changing the structure of output (intervals rearrangements) so that the programs now work as described. If you used earlier changing structure with the arguments then all the results must be re-considered (re-done) with fixed version 0.18.0.0. The author ask for pardon because of such longlasted issue. Improved documentation. Added examples of two new sets of values of durations that are available as the ready-to-use file from the persistent link: [open here](#). Fixed issues with string interpreter (see the appropriate information) and added the possibility to adjust splitting.

Changes in version 0.19.0.0

The new version adds support for additional properties, as well as multiple properties mode for distributionTextG. The theoretical part of this document has been improved. Fixed some code bugs. Switched in calculations of the average values and standard deviation from population to sample versions (now the standard deviation is slightly higher). Fixed issues with multiple variations of the text mode if it is not used interactively.

Changes in version 0.20.0.0

In the new version, a significant change in functionality was made, a syllable-as-a-whole (SaaW) mode was added, which allows taking into account not only the durations of phonemes or phonetic phenomena, but also setting syllable weights, which allows in primarily

distinguish stressed and unstressed syllables, as well as take into account a wide range of prosodic factors, including logical stresses, intonation, the general nature of pronunciation, etc., and can also be used to create syllabotonic poetry, musical works, etc. Also, this generalization allows analyzing a wider range of languages as phonetic ones. Fixed errors in segmentation into syllables and words. Now words with hyphens and apostrophes are segmented in a better, more appropriate way (to the needs of the approach and in general). Fixed an incomplete set of permutations for encoding-decoding the 10th word that could cause incorrect operation. For this purpose, the transition from the decimal to the hexadecimal numbering system was carried out. In general, all changes lead to the fact that the output of programs may partially change, and therefore, if accuracy in calculations is required, then the results must be recalculated

Working with the program lineVariantsG3

Verify that the folder (directory) where cabal installed the executables of the programs is available for search in the PATH environment variable.

The new version has two main modes of operation:

1. with syllable-as-one-whole (SaaW), when each syllable is considered as one whole, as a basic unit, with the possibility of specifying a numerical characteristic in the form of a number of type Double; it applies if:
 - among the property designations there is at least one with the letter 'a' in writing;
 - among the arguments of the command line consecutively (outside the groups +m ...-m and +a -a) enter "+s" followed by a natural number, not less than 1, which indicates how many different sets of pairs "syllable-number" will be set by the user for the operation of the mode;
2. normal (was the only one possible before this new version), phonetic-phenomenon-as-a-whole (PhoPaaW) mode, when the durations of phonetic phenomena (usually phonemes) are set in advance, they become constant during the program call.

In turn, each mode can also be used in two other modes of operation (a total of $2 \cdot 2 = 4$ options for combinations of modes):

- with one property (normal mode);
- with several (no more than five different) properties.

The latter is used if the command line arguments include a group entered by delimiters +m <property type1> <numeric arguments1> <property type2> <numeric arguments2> <property type3> <numeric arguments3> <property type4> <numeric arguments4> <property type5> <numeric arguments5> -m.

More about this at relevant section (see link above). The operation of the program in this mode is described later in a separate section.

To work in the single property mode, enter the command at the command prompt (or terminal):

```
lineVariantsG3 <the first argument> [<WX argument> <whether to print property value(s)>
<whether to fix the last word> <stay in place> <whether to use recursive interactive mode>]
[<whether to use the multiple sources mode>] <numerical arguments> < property type>
<Ukrainian text>
```

everything here and further in the single line or using the terminal line hyphenation, or:

```
lineVariantsG3 <the first argument> [<WX argument> <whether to print property value(s)>
<whether to fix the last word> <stay in place > <whether to use recursive
interactive mode>] <numeric arguments > <property type >
[<whether to use the multiple sources mode>] <Ukrainian text>
<somewhere in the middle arguments as a single group: restrictions>
```

and press Enter. Additionally, you can set the interactive mode, about what see below for more details .

If you don't specify groups in square brackets, you'll see something like this:

```
lineVariantsG3 10.0_1.2 уу2 садок вишневий коло хати хруці над вишнями гудуть
```

(Ukrainian text entered in the end of the command line arguments - an excerpt from a famous poem by Taras Shevchenko; in general this line is the entered command). Here and further is cited by: [77]

```
садок колохати хруці гудуть надвишнями вишневий
```

(in general, there may be several such variants that form one group, as well as several such groups; all groups follow one another from top to bottom in descending order of the final value of the property, which maximizes the selected property for given intervals)

- 3.6562 (value of selected property before applying interval conversion)
3.6562 (the value of the selected property after the interval conversion, the final value of the property for this line).

Note that the text may (and probably will not) be written the way it is spelled according to spelling and punctuation rules, but you can read it and try to understand. By modifying the first arguments entered, you will (most likely) get other output, the same obviously, it also applies to the Ukrainian text. Too long text will be reduced to a volume that you could understand (perhaps after previously mentioned training) without effort.

Try to evaluate by reading the variant how it is suitable.

NOTE: Also keep in mind that in single property mode, numeric arguments precede the notation of a property, and in the case of multiple properties (see below) on the contrary - the designation of the property begins a set of numerical arguments that relate to it, if any (otherwise default values are used, which are just the same to the search for the maximum element).

Ukrainian information messages

In order for the program to display informational messages in Ukrainian (it displays in English by default), it is required to specify '+u' as one of the command line arguments somewhere outside the option groups, for example, at the beginning. You can also customize the alias for this version of the program, as for other options, if you like. Refer to the documentation for details command shells.

More complex usage

Numeric arguments, if specified, have the following meaning.

The first numeric argument is the number of groups with the same maximum property value (in descending order) that will be output on the screen as a result. If you specify a larger number than there is at all, then all possible results are displayed that satisfy all other conditions. If no numeric arguments are specified, it is considered equal to 1.

The second numeric argument is the number of intervals into which the interval between the minimum and maximum value of the property for this line. If not specified, it is considered equal to 1. A value of 0 does not allow other numeric arguments to further change the result of the work of the program.

All subsequent numeric arguments (if specified, otherwise no permutations occur) are interval numbers that will be swapped with the maximum number. This allows you to change the structure of the data that is displayed as a result of the program and see the internal (not maximum) items. For example, the numeric arguments 2 6 1 4 (in this order) will mean that during execution the program will return 2 groups of elements with the maximum values of the property (the largest and the next largest ones) obtained after permutations of the intervals; the interval between the maximum and minimum value of the property will be divided into 6 equal intervals, thus elements that are in the first and fourth, counting from the minimum (interval number 1) will be moved to the largest one; the command will display 2 groups of elements. Values that were in the maximum interval will be moved to the interval with the lowest number among those that are moved to the maximum one. Thus, at an output these values will be deduced at the latest.

Parameter +l (+bl) and its usage

NOTE: If there were no +l, +bl, +i, +y ... command line arguments, then each output block will have 2 numbers in square brackets displayed - the initial value of the property (without moving the intervals) and the value after moving. If there was (at least) one of these (groups) of characters - the value of the properties will not be printed.

It should also be remembered that:

+bl = +b +l

(this is just a reduction in the use of both parameters at once, instead of 5 characters you only need to enter 3).

If you also specify +f ... or +i, then this parameter may not be specified (it will be applied automatically), instead, if you want, you can specify an additional +b instead.

Parameter +b (+bl) and its usage

If somewhere among the arguments of the command line specify an argument in the form of +b (or +bl), the program will preserve, when outputting and analyzing, the last word in the line in its place - it is very convenient when you need, having a rhyme, to pick up other words. If not specified, then all words will be moved (if necessary). The operation of the parameter is actually implemented as an additional constraint, see the following sections. You can also set additional constraints.

+bl = +b +l

(this is just a reduction in the use of both parameters at once, instead of 5 characters you only need to enter 3).
About the use of other parameters a little later.

Multiple properties mode (+m ...-m)

If you specify a group of arguments selected by the +m and -m delimiters from the command line arguments so that the argument group is selected +a and -a delimiters were not inside this, and vice versa (so that they do not intersect), then the program will work in the multiple properties mode. The values of the properties will not be displayed on the screen, instead it is possible to set no more than four different properties and to each of them to specify arguments (see: More complex usage). The program will then find variants that meet each of these conditions, and then display only those variants that are found in all selected and given properties with parameters. Numerical arguments that stand after the property designation and precede the next denote a property related to that property. If numeric arguments are omitted, the default values are used (in fact, this is equivalent to simply searching for maximum property values).

In general, this is the more comprehensive use of this program.

Try, for example, to specify:

```
lineVariantsG3 +m 02y 3 03y 3 y0 10 -m +bl <Ukrainian text>.
```

Interactive mode (+i) and its usage

Interactive mode (additional extended user interaction, in addition to the required) is enabled and set accordingly by the command line argument +i, which can be placed anywhere in the command line. In this case, the program displays not just lines that satisfy all conditions, but for each line also displays its sequence number (starting with 1) in order of increasing the 'weakness' of all conditions (the higher the number in the general case, the more likely the weaker effect is of given conditions, although this is not always the case - in particular when you need to withdraw only one group). The program then asks what the choice is waits for the option number entered by the user. Then returns that option without a number.

It looks something like this:

```
lineVariantsG3 +i +m 02y 10 03y 10 y0 35 -m +bl садок вишневий коло хати хрущі над вишнями гудуть
0
```

Please, check whether the line below corresponds and is consistent with the constraints you have specified between the +a and -a options. Check also whether you have specified the "+b" or "+bl" option(s). If it is inconsistent then enter further "n", press Enter and then run the program again with better arguments.

If the line is consistent with your input between +a and -a then just press Enter to proceed further.

```
садок вишневий колохати хрущі надвишнями гудуть
```

```
1 вишневий колохати садок хрущчі надвишнями гудуть
2 вишневий хрущчі колохати садок надвишнями гудуть
3 вишневий хрущчі садок колохати надвишнями гудуть
4 вишневий хрущчі садок надвишнями колохати гудуть
5 садок вишневий колохати хрущчі надвишнями гудуть
6 хрущчі вишневий колохати садок надвишнями гудуть
7 садок вишневий хрущчі колохати надвишнями гудуть
8 хрущчі вишневий садок колохати надвишнями гудуть
9 садок вишневий хрущчі надвишнями колохати гудуть
10 хрущчі вишневий садок надвишнями колохати гудуть
11 хрущчі колохати вишневий садок надвишнями гудуть
12 садок хрущчі вишневий колохати надвишнями гудуть
13 надвишнями садок вишневий хрущчі колохати гудуть
14 садок надвишнями вишневий хрущчі колохати гудуть
15 садок хрущчі вишневий надвишнями колохати гудуть
16 колохати надвишнями садок вишневий хрущчі гудуть
17 надвишнями колохати садок вишневий хрущчі гудуть
```

- 18 надвишньами колохати хрушчы́ вишневий садок гудуть
- 19 надвишньами хрушчы́ колохати вишневий садок гудуть
- 20 садок надвишньами хрушчы́ вишневий колохати гудуть
- 21 хрушчы́ надвишньами садок вишневий колохати гудуть
- 22 садок хрушчы́ надвишньами вишневий колохати гудуть
- 23 хрушчы́ садок надвишньами вишневий колохати гудуть
- 24 колохати надвишньами садок хрушчы́ вишневий гудуть
- 25 колохати хрушчы́ надвишньами садок вишневий гудуть
- 26 надвишньами колохати садок хрушчы́ вишневий гудуть
- 27 надвишньами колохати хрушчы́ садок вишневий гудуть
- 28 хрушчы́ колохати надвишньами садок вишневий гудуть
- 29 надвишньами садок колохати хрушчы́ вишневий гудуть
- 30 надвишньами хрушчы́ колохати садок вишневий гудуть
- 31 садок надвишньами колохати хрушчы́ вишневий гудуть
- 32 хрушчы́ надвишньами колохати садок вишневий гудуть
- 33 надвишньами садок хрушчы́ колохати вишневий гудуть
- 34 надвишньами хрушчы́ садок колохати вишневий гудуть
- 35 садок надвишньами хрушчы́ колохати вишневий гудуть
- 36 хрушчы́ надвишньами садок колохати вишневий гудуть
- 37 хрушчы́ садок надвишньами колохати вишневий гудуть

Please, specify the variant which you would like to become the resulting string by its number.

5

садок вишневий колохати хрушчы́ надвишньами гудуть

Interactive mode of writing a line to a file (+f ...)

If you specify a group of three arguments as +f <path to record file>, then in the specified path to the text file specified, if possible, the final result of the program will be appended, except that it will be displayed as in the usual interactive mode on the screen. This command line arguments group can be anywhere between the command line arguments of the program call, but should not be contained inside other arguments of the form +a ... -a, +m ... -m, etc.

The result can be something like this:

```
lineVariantsG3 +f hello.txt +bl +m 02y 10 0y 10 y0 40 -m садок вишневий коло хати хрущі над \
    вишнями гудуть
0
```

Please, check whether the line below corresponds and is consistent with the constraints you have specified between the +a and -a options. Check also whether you have specified the "+b" or "+bl" option(s). If it is inconsistent then enter further "n", press Enter and then run the program again with better arguments.

If the line is consistent with your input between +a and -a then just press Enter to proceed further.

```
садок вишневий колохати хрущі надвишнями гудуть
```

```
1 вишневий хрущчі колохати садок надвишнями гудуть
2 колохати вишневий надвишнями садок хрущчі гудуть
3 садок вишневий колохати хрущчі надвишнями гудуть
4 хрущчі вишневий колохати садок надвишнями гудуть
5 надвишнями колохати вишневий хрущчі садог гудуть
6 хрущчі колохати вишневий садок надвишнями гудуть
7 надвишнями хрущчі вишневий колохати садог гудуть
8 садок хрущчі вишневий колохати надвишнями гудуть
9 садок надвишнями вишневий хрущчі колохати гудуть
10 надвишнями колохати хрущчі вишневий садог гудуть
```

```

11  хрущчі колохати садок вишневий надвишньами гудуть
12  надвишньами хрущчі колохати вишневий садок гудуть
13  садок надвишньами хрущчі вишневий колохати гудуть
14  хрущчі надвишньами садок вишневий колохати гудуть
15  колохати надвишньами садок хрущчі вишневий гудуть
16  колохати хрущчі надвишньами садок вишневий гудуть
17  надвишньами колохати садок хрущчі вишневий гудуть
18  надвишньами колохати хрущчі садок вишневий гудуть
19  надвишньами садок колохати хрущчі вишневий гудуть
20  надвишньами хрущчі колохати садок вишневий гудуть
21  садок надвишньами хрущчі колохати вишневий гудуть
22  хрущчі надвишньами садок колохати вишневий гудуть

```

Please, specify the variant which you would like to become the resulting string by its number.

3

```
садок вишневий колохати хрущчі надвишньами гудуть
```

This end line in the program output will also be appended to the file with the specified name, if possible for this user.

If you wish, you can run the command again with new text and / or new arguments. If similarly the same file is specified, then the new result will be appended further to the same file. This makes it possible to apply this program consistently, write or rewrite texts (e.g. poems).

Mode of simultaneous possible variations of the text

Starting with version 0.3.0.0, the ability to process several variations of text at once has been added, in particular those that deals with synonyms, paraphrases, etc.

To do this, use the following special construction instead of plain text as extreme arguments:

```
{<variant1 of the Ukrainian text> / <variant2 of the Ukrainian text> / ... /
<variantN of the Ukrainian text>}
```

everything at the single line with at least two variants inside curly braces. These options will be worked out in turn each in particular during one call of the program, and you will select one of the variants (possibly the empty one). In the end there will be an opportunity to choose among these pre-prepared versions of the only one, single final variant, which will be the result (and accordingly, for example, will be displayed and written to a file if provided by command line arguments).

Please note that the program in this mode provides processing of each of the possible combinations of variations, and therefore, if you specify too many of them (for example, 3 variations on one word and 4 on another will create $3 * 4 = 12$ variations which will be consecutively processed) the execution time of the program can be longer than expected until you get the final result.

ATTENTION: starting from the version 0.19.0.0:

If you do not use interactive mode (either write to file mode or recursive mode), the options will be displayed in groups, with an empty line between them.

In interactive mode (either recursive or write to file mode), the options will be analyzed for each combination in turn, and the final will be common among all.

Recursive mode of the work (“+r”)

Starting from the version 0.9.0.0 you can execute the program in the recursive interactive mode. For this, you need to run the command with the parameter “+r”, e. g. at the beginning after the first argument. In such a case the program will be executed recursively, proposing to end the recursion on every step. The result of the last step will be the overall result of the program execution.

This mode is incompatible with constraints (because the constraints do not have the proper expected meaning and begin to ‘shift’ from the needed parts of the text to the other ones), that is why they should not be used simultaneously, this mode can be the alternative to the latter ones. While execution in the recursive mode there is no possibility to change the call parameters for the properties etc., therefore, choose them wisely.

The text changes in this mode are specified by the so called ‘interpreter string’, i. e. the textual input that in the arithmetic expression – either in a number or division expression encodes the following actions of the program.

- If on the interpreter string the two-digit number is entered then the first digit the program tries to interpret as an index of the first word (starting counting from 1) to which the change is applied, the second digit is interpreted as a number of the words that are concatenated further including the first specified one. Afterwards, the program (if it is possible to do so without an error) works with the newly generated text. For example, the “12” means that the program will concatenate the first word (digit “1”) with

the following ones in the quantity of 2 together (digit "2") this leads to concatenation of the first two words. "34" means that the program tries to concatenate consequently 4 words starting from the third one. If it is impossible the program will execute the previous stage again with prompt to input the interpreter string again.

- If the three- or multi-digit number is specified then all the digits that are not equal to 0 the program tries to interpret as the indices of the words that are needed to be concatenated in the order of the digits written down in the interpreter string.
- If the digit (greater than 0 and not greater than the number of words on the line) precedes the '/' and then is followed by integer number with sign. Then the first digit before the '/' sign means the index of the words (counting starts from 1) that is splitted into 2 parts, the second number is the quantity of the symbols that the program counts from the beginning of the word (in case of positive integer) to the right or from the end of the word (in case of negative integer) to the left to split it into two parts (the positive case is similar to the work of the standard function 'splitAt'). Then the program, if the specified string was successfully interpreted in such a way, will split the specified word into two parts (one of which can be an empty string) and will work with the text in which the specified word is substituted with these two new words (or just the first of them, if they are not empty and the number of the words in the line is already equal to 7 at the moment of splitting or 10 in case of incomplete permutations set). For example, the interpreter string "1/5" will split the "садоквишневий" if it is the first word in the text into "садок вишневий" (counting from the beginning the 5 symbols) and will work with the new text further. Starting with version 0.18.0.0, there has been added a check for long word concatenations. If after '/' enter a number whose module is greater than 7 (for example, 9 or -15), then it will first display the word, which must be splitted into two parts according to the value you entered (as just described), adding the "?: " in front of it; if the split is done correctly, then you need to press Enter. If not, then you need to specify the word (number 1 or 2) and through a space ("shift" space left or right to) N (must be greater than 0), all this is entered immediately in a new line, and then again there will be a query "?: " type. The program will repeat the splitting and adjusting until you simply press Enter. This allows you not to count more than 8 characters in a concatenation complex, trying to get the split in the right place for long combinations, and quickly determine "by eye", approximately, and then adjust. This option has been added for convenience.

WX argument

If among properties you use “w” or “x” series (or both ones) then you can specify for them another argument that must begin with “+x” with the next written down two positive Double numbers connected by the ‘_’ symbol (underline symbol). For example, +x 2.345_0.45676237876. If this argument is not specified explicitly then the default one +x 2.0_0.125 is used.

Please, note that starting from the version 0.16.0.0 now there is a space between the +x and numbers. Earlier, there has not been.

The first number is used as a factor (multiplier or divisor in case of non-suitability) and it mostly influences the property value, it deals with the most important syllables in the rhythmic group; on the other hand, the second one is used either just for increasing of the property value if the less highlighted syllable durations in the rhythmic group (corresponds to ‘w’ series) or also for more complex behaviour (‘x’ series).

More details can be found in the section Types of properties.

Multiple sources usage mode

If among the command line arguments before the Ukrainian text (or instead of one) or after it one specifies a group of arguments in the

```
+t {two-digits number} ... ^t
```

frame then the program will work using the multiple sources mode. Here, instead of ellipsis you can specify also arguments.

The program’s behaviour differs for the cases of pairwise permutations (see: Pairwise permutations as a variant of the universal set of permutations, it is set using the command line arguments «+p [1 or 2]») and the full universal set of permutations.

In the first case, after the +t the program waits for one of the numbers of: 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 70, 71, 80, 81, 90, 91, where 0 as the second digit means that the program won’t have concatenate the lines from each source into one singular line before its analysis and processments, and the 1 there means that the program will at first concatenate all the lines of each source into the one single line and only then it will apply the splitting into the parts and other processments specified by other arguments. The first digit (if not equal to 1) means the number of the words in the lines (except the last one in every source) that the program will analyse consequently. If the first digit equals to 1, then this is equivalent to the 10 (and this is the maximum number of the words in this case).

In the second case, after the +t the program expects one of the following numbers: 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71. Their meaning is the same as for the first case.

At the place of the ellipsis you can specify the paths to the files with Ukrainian texts which is used for the analysis. If they are left empty then the program will prompt every time the new line in the non-terminating loop (interruptible by the user) and, afterwards, will process it accordingly to all the other arguments. You can also in such a case just specify +t (with the next two-digits number).

Just try, e. g.:

```
lineVariantsG3 +r 3 w02 +t 70 "sajok.txt" "other_poem.txt" "just_text.txt" ^t
lineVariantsG3 +r 3 w02 +t 70
lineVariantsG3 +r 3 w02 +f "fileForSaving.txt" +t 50
```

where it is expected that the specified files do exist and can be read (the program ignores the files it cannot read).

This mode is also incompatible with the constraints (+a ... -a), instead you should use the recursive mode.

Please, note that in the new version of the program (starting from the 0.16.0.0) the * sign is changed to the + sign, besides there is a space between the first argument and the number.

Additional properties

Starting with version 0.19.0.0 of the phonetic-languages-simplified-examples-array package, it is possible to use special additional properties to improve the output for many properties. All of them are set by using among the property letters also the English letter 'G' (it was previously reserved) and a special format of the numerical argument to it. These additional properties have a special meaning, which differs significantly from the general scheme of calculations.

Poetic Line

Consider an example that shows that it is appropriate to use such additional properties.

If you save a text file with text information from the page 56.csv named 56.csv in the working directory (folder) (where you are in the terminal or console) and execute the command:

```
lineVariantsG3 +d 56.csv 11 HI32 весна уже прийшла і спів пташок лунає звідусіль
0
```


пташок прийшла уже весна іспів лунає звідусіль
Infinity
Infinity
лунає пташок уже іспів прийшла весна звідусіль
Infinity
Infinity
лунає пташок весна іспів прийшла уже звідусіль
Infinity
Infinity
лунає прийшла уже весна пташок іспів звідусіль
Infinity
Infinity
звідусіль пташок уже іспів прийшла лунає весна
Infinity
Infinity
звідусіль уже прийшла лунає іспів пташок весна
Infinity
Infinity
прийшла лунає звідусіль уже іспів пташок весна
Infinity
Infinity
уже прийшла звідусіль іспів весна пташок лунає
Infinity
Infinity
уже весна звідусіль іспів прийшла пташок лунає
Infinity
Infinity
весна пташок звідусіль прийшла іспів уже лунає
Infinity

Infinity

пташок прийшла звідусіль іспів уже весна лунає

Infinity

Infinity

звідусіль весна пташок прийшла уже іспів лунає

Infinity

Infinity

весна пташок звідусіль прийшла уже іспів лунає

Infinity

Infinity

прийшла пташок звідусіль уже іспів весна лунає

Infinity

Infinity

уже пташок звідусіль іспів весна прийшла лунає

Infinity

Infinity

уже весна звідусіль іспів пташок прийшла лунає

Infinity

Infinity

пташок іспів весна звідусіль уже прийшла лунає

Infinity

Infinity

пташок іспів весна звідусіль лунає прийшла уже

Infinity

Infinity

звідусіль пташок весна іспів прийшла лунає уже

Infinity

Infinity

звідусіль весна пташок прийшла лунає іспів уже

Infinity

Infinity

пташок прийшла лунає звідусіль весна іспів уже

Infinity

Infinity

весна звідусіль уже лунає прийшла іспів пташок

Infinity

Infinity

уже лунає весна звідусіль прийшла іспів пташок

Infinity

Infinity

прийшла іспів уже звідусіль весна лунає пташок

Infinity

Infinity

іспів прийшла звідусіль весна уже лунає пташок

Infinity

Infinity

лунає пташок уже весна звідусіль іспів прийшла

Infinity

Infinity

пташок іспів весна звідусіль уже лунає прийшла

Infinity

Infinity

уже весна звідусіль іспів пташок лунає прийшла

Infinity

Infinity

весна іспів уже звідусіль лунає пташок прийшла

Infinity

Infinity

them on the screen.

But if you want to "manually" select option 4, which gave such a result, you will need to do it manually, or use additional programming, which may be inconvenient for the end user, who does not program. Instead, if you could take all the values that are, for example, more than 1 (or 1000), you get exactly the data as given.

To go further it is to be said that starting from the version 0.19.0.0 you can do this the following way:

```
lineVariantsG3 +d 56.csv +m GHI32 01 HI32 20000 -m весна  уже прийшла і спів пташок лунає звідусіль
0
весна звідусіль уже лунає прийшла іспів пташок
весна звідусіль уже лунає пташок прийшла іспів
весна уже звідусіль прийшла іспів лунає пташок
весна уже звідусіль пташок іспів лунає прийшла
весна іспів звідусіль уже пташок лунає прийшла
весна пташоґ звідусіль прийшла уже іспів лунає
весна пташоґ звідусіль прийшла іспів уже лунає
весна іспів уже звідусіль лунає пташок прийшла
весна лунає уже прийшла пташоґ звідусіль іспів
весна лунає уже пташок прийшла звідусіль іспів
звідусіль весна пташок прийшла лунає іспів уже
звідусіль весна пташок прийшла уже іспів лунає
уже весна звідусіль пташок іспів лунає прийшла
уже весна звідусіль іспів пташок лунає прийшла
уже весна звідусіль іспів прийшла пташок лунає
уже весна звідусіль іспів пташок прийшла лунає
звідусіль прийшла весна уже пташок лунає іспів
звідусіль пташок весна іспів прийшла лунає уже
іспів звідусіль весна уже прийшла лунає пташок
іспів звідусіль весна уже пташок лунає прийшла
уже лунає весна звідусіль прийшла іспів пташок
уже лунає весна звідусіль пташок прийшла іспів
```


пташок іспів весна звідусьіль лунаїе прийшла уже
пташок іспів весна звідусьіль уже лунаїе прийшла
пташок іспів весна звідусьіль уже прийшла лунаїе
іспів лунаїе весна прийшла пташоґ звідусьіль уже
іспів лунаїе весна пташок прийшла звідусьіль уже
іспів уже весна прийшла пташоґ звідусьіль лунаїе
лунаїе пташок весна іспів прийшла уже звідусьіль
іспів звідусьіль уже весна прийшла лунаїе пташок
іспів звідусьіль уже весна пташок лунаїе прийшла
іспів прийшла звідусьіль весна уже лунаїе пташок
іспів пташоґ звідусьіль весна уже лунаїе прийшла
лунаїе пташок уже весна звідусьіль іспів прийшла
лунаїе прийшла уже весна пташок іспів звідусьіль
пташок прийшла уже весна іспів лунаїе звідусьіль
уже прийшла звідусьіль іспів весна пташок лунаїе
уже пташоґ звідусьіль іспів весна прийшла лунаїе
пташок прийшла лунаїе звідусьіль весна іспів уже
прийшла іспів уже звідусьіль весна лунаїе пташок
іспів пташок уже прийшла весна звідусьіль лунаїе
прийшла пташоґ звідусьіль уже іспів весна лунаїе
пташок прийшла звідусьіль іспів уже весна лунаїе
пташок прийшла лунаїе звідусьіль уже весна іспів
лунаїе пташок уже іспів прийшла весна звідусьіль
звідусьіль уже прийшла лунаїе іспів пташок весна
звідусьіль пташок уже іспів прийшла лунаїе весна
прийшла лунаїе звідусьіль уже іспів пташок весна
іспів лунаїе уже прийшла пташоґ звідусьіль весна
іспів лунаїе уже пташок прийшла звідусьіль весна

And in such a case, the rule that in the output the higher variants have the greater value of the property is not applicable.

lineVariantsG3 +r +d 56.csv +u 01 GHI32 весна уже прийшла і спів пташок лунає звідусіль

0
1 пташок прийшла уже весна іспів лунаїе звідусіль
2 лунаїе пташок уже іспів прийшла весна звідусіль
3 лунаїе пташок весна іспів прийшла уже звідусіль
4 лунаїе прийшла уже весна пташок іспів звідусіль
5 звідусіль пташок уже іспів прийшла лунаїе весна
6 іспів лунаїе уже пташок прийшла звідусіль весна
7 звідусіль уже прийшла лунаїе іспів пташок весна
8 прийшла лунаїе звідусіль уже іспів пташок весна
9 іспів лунаїе уже прийшла пташоґ звідусіль весна
10 уже прийшла звідусіль іспів весна пташок лунаїе
11 іспів уже весна прийшла пташоґ звідусіль лунаїе
12 уже весна звідусіль іспів прийшла пташок лунаїе
13 весна пташоґ звідусіль прийшла іспів уже лунаїе
14 пташок прийшла звідусіль іспів уже весна лунаїе
15 звідусіль весна пташок прийшла уже іспів лунаїе
16 весна пташоґ звідусіль прийшла уже іспів лунаїе
17 прийшла пташоґ звідусіль уже іспів весна лунаїе
18 уже пташоґ звідусіль іспів весна прийшла лунаїе
19 уже весна звідусіль іспів пташок прийшла лунаїе
20 пташок іспів весна звідусіль уже прийшла лунаїе
21 іспів пташок уже прийшла весна звідусіль лунаїе
22 пташок іспів весна звідусіль лунаїе прийшла уже
23 звідусіль пташок весна іспів прийшла лунаїе уже
24 іспів лунаїе весна пташок прийшла звідусіль уже
25 іспів лунаїе весна прийшла пташоґ звідусіль уже
26 звідусіль весна пташок прийшла лунаїе іспів уже
27 пташок прийшла лунаїе звідусіль весна іспів уже

28 весна звідусьіль уже лунайє прийшла іспів пташок
29 уже лунайє весна звідусьіль прийшла іспів пташок
30 весна уже звідусьіль прийшла іспів лунайє пташок
31 прийшла іспів уже звідусьіль весна лунайє пташок
32 іспів звідусьіль весна уже прийшла лунайє пташок
33 іспів звідусьіль уже весна прийшла лунайє пташок
34 іспів прийшла звідусьіль весна уже лунайє пташок
35 весна уже звідусьіль пташок іспів лунайє прийшла
36 уже весна звідусьіль пташок іспів лунайє прийшла
37 лунайє пташок уже весна звідусьіль іспів прийшла
38 пташок іспів весна звідусьіль уже лунайє прийшла
39 уже весна звідусьіль іспів пташок лунайє прийшла
40 весна іспів уже звідусьіль лунайє пташок прийшла
41 весна іспів звідусьіль уже пташок лунайє прийшла
42 іспів звідусьіль весна уже пташок лунайє прийшла
43 іспів звідусьіль уже весна пташок лунайє прийшла
44 іспів пташоґ звідусьіль весна уже лунайє прийшла
45 весна лунайє уже пташок прийшла звідусьіль іспів
46 весна звідусьіль уже лунайє пташок прийшла іспів
47 уже лунайє весна звідусьіль пташок прийшла іспів
48 пташок прийшла лунайє звідусьіль уже весна іспів
49 весна лунайє уже прийшла пташоґ звідусьіль іспів
50 звідусьіль прийшла весна уже пташок лунайє іспів

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

23

1 пташок прийшлаужевесна іспів лунаїе звідусьіль
 2 лунаїе прийшлаужевесна пташок іспів звідусьіль

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1
 Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

021

1 лунаїе прийшлаужевеснапташок іспів звідусьіль

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1
 Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

лунаїе прийшлаужевеснапташок іспів звідусьіль

So you can suggest the following line:

Лунає: "Прийшла уже весна пташок", – і спів звідусіль (It should be added that a certain level of figurative thinking and associativity is required to understand such a line).

If you look at the initial output, it was line 4. But it could be 50, the program does not organize the data in this mode, but instead of 5040 options in the first stage offers only 50 (you do not need to read all, generally speaking), and then much less.

The thoughtful reader will notice that the line is ambiguous, because it is impossible to say unequivocally whose singing and whose direct speech are. Generally speaking, if you stick only to phonetic properties, it increases the ability to compose such lines, but you can not strictly stick to phonetic properties, then at a certain stage you can choose the option that is best from a semantic point of view.

Another example.

lineVariantsG3 +r +d 56.csv +u 01 GHK32 весна { уже / вже } прийшла і спів пташок { лунає / звучить }
звідусіль

```

0
1   пташоґ звучить іспів вжеприйшла весна звідусіль
2   пташок іспів звучить вжеприйшла весна звідусіль
3   пташоґ звучить вжеприйшла іспів весна звідусіль
4   пташоґ звучить вжеприйшла весна іспів звідусіль
5   пташок вжеприйшла весна звучить іспів звідусіль
6   пташок вжеприйшла звучидь звідусіль іспів весна
7   звідусіль весна вжеприйшла іспів пташоґ звучить
8   звідусіль весна пташок вжеприйшла іспів звучить
9   пташок вжеприйшла весна звідусіль іспів звучить
10  звідусіль пташок весна звучить іспів вжеприйшла
11  пташоґ звучидь звідусіль іспів весна вжеприйшла
12  звідусіль весна звучить вжеприйшла іспів пташок
13  звідусіль весна звучить іспів вжеприйшла пташок
14  звучить пташоґ звідусіль вжеприйшла весна іспів
15  пташоґ звідусіль звучить вжеприйшла весна іспів
16  пташок вжеприйшла весна звучидь звідусіль іспів
17  пташоґ звідусіль вжеприйшла звучить весна іспів

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

012

```

1   весна звідусіль пташоґзвучить вжеприйшла іспів
2   пташоґзвучить іспів вжеприйшла весна звідусіль

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

021

(/ Ви вказали властивості(ість) та діапазон(и) для них такі, що для даних слів та їх сполучень варіантів немає. Спробуйте змінити параметри виклику програми (бібліотеки) /)

```

1   пташок весна іспів вжеприйшла лунаїе звідусьіль
2   пташок іспів весна вжеприйшла лунаїе звідусьіль
3   вжеприйшла лунаїе пташок іспів весна звідусьіль
4   лунаїе весна вжеприйшла іспів пташоґ звідусьіль
5   пташок вжеприйшла весна лунаїе іспів звідусьіль
6   вжеприйшла лунаїе пташок весна іспів звідусьіль
7   звідусьіль лунаїе пташок іспів вжеприйшла весна
8   пташоґ звідусьіль вжеприйшла лунаїе іспів весна
9   пташок вжеприйшла лунаїе звідусьіль іспів весна
10  звідусьіль весна пташок вжеприйшла іспів лунаїе
11  пташок вжеприйшла весна звідусьіль іспів лунаїе
12  звідусьіль лунаїе пташок іспів весна вжеприйшла
13  пташоґ звідусьіль лунаїе іспів весна вжеприйшла
14  лунаїе звідусьіль вжеприйшла іспів весна пташок
15  вжеприйшла звідусьіль лунаїе весна іспів пташок
16  звідусьіль лунаїе пташок весна вжеприйшла іспів
17  пташоґ звідусьіль вжеприйшла весна лунаїе іспів
18  пташоґ звідусьіль вжеприйшла лунаїе весна іспів

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

4

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

```
лунайє весна вжеприйшла іспів пташоґ звідусьіль
1   пташоґ звучить прийшла весна іспів уже звідусьіль
2   уже пташок прийшла весна звучить іспів звідусьіль
3   пташоґ звучить прийшла весна уже іспів звідусьіль
4   звучить пташок прийшла уже весна іспів звідусьіль
5   весна іспів уже пташоґ звучить прийшла звідусьіль
6   прийшла звідусьіль пташок іспів звучить уже весна
7   пташоґ звідусьіль прийшла іспів звучить уже весна
8   пташок прийшла іспів звідусьіль звучить уже весна
9   пташок іспів прийшла звідусьіль звучить уже весна
10  пташок іспів звідусьіль прийшла уже звучить весна
11  прийшла пташоґ звідусьіль іспів уже звучить весна
12  пташоґ звучидь звідусьіль іспів уже прийшла весна
13  прийшла звучидь звідусьіль іспів уже пташок весна
14  звучить прийшла пташок уже іспів звідусьіль весна
15  пташоґ звідусьіль звучить уже прийшла іспів весна
16  уже звідусьіль весна прийшла іспів пташоґ звучить
17  прийшла весна звідусьіль уже іспів пташоґ звучить
18  прийшла уже звідусьіль весна іспів пташоґ звучить
19  прийшла уже звідусьіль іспів весна пташоґ звучить
20  весна іспів звідусьіль прийшла уже пташоґ звучить
21  пташок іспів прийшла весна звідусьіль уже звучить
22  пташоґ звідусьіль прийшла іспів весна уже звучить
23  прийшла звідусьіль пташок іспів весна уже звучить
```

24 прийшла пташоґ звыідусьіль іспьів уже весна звучить
25 весна прийшла пташок уже іспьів звыідусьіль звучить
26 прийшла пташоґ звыідусьіль уже іспьів весна звучить
27 прийшла весна уже пташоґ звыідусьіль іспьів звучить
28 прийшла уже весна пташоґ звыідусьіль іспьів звучить
29 прийшла уже пташоґ звыідусьіль весна іспьів звучить
30 звыідусьіль пташок уже весна іспьів прийшла звучить
31 весна іспьів звыідусьіль пташок уже прийшла звучить
32 весна звыідусьіль звучить пташок іспьів прийшла уже
33 звыідусьіль пташок весна звучить іспьів прийшла уже
34 пташоґ звучидь звыідусьіль іспьів весна прийшла уже
35 пташок іспьів прийшла звыідусьіль звучить весна уже
36 прийшла звучидь звыідусьіль іспьів весна пташок уже
37 прийшла весна звучить пташоґ звыідусьіль іспьів уже
38 прийшла звучить весна пташоґ звыідусьіль іспьів уже
39 прийшла звучить пташоґ звыідусьіль весна іспьів уже
40 прийшла пташоґ звучидь звыідусьіль іспьів весна уже
41 прийшла пташоґ звыідусьіль весна іспьів звучить уже
42 прийшла пташоґ звыідусьіль іспьів весна звучить уже
43 прийшла звыідусьіль пташок іспьів звучить весна уже
44 пташоґ звыідусьіль звучить весна прийшла іспьів уже
45 пташоґ звучить весна прийшла звыідусьіль іспьів уже
46 пташоґ звыідусьіль прийшла іспьів звучить весна уже
47 пташок прийшла іспьів звыідусьіль звучить весна уже
48 звучить весна уже прийшла звыідусьіль іспьів пташок
49 звыідусьіль звучить уже весна прийшла іспьів пташок
50 звучить уже весна прийшла звыідусьіль іспьів пташок
51 звыідусьіль уже звучить весна прийшла іспьів пташок
52 прийшла весна звучидь звыідусьіль уже іспьів пташок

53 прийшла звучить весна звьідусьіль уже іспьів пташок
54 прийшла звучидь звьідусьіль уже іспьів весна пташок
55 прийшла звучить уже звьідусьіль іспьів весна пташок
56 прийшла весна уже звучидь звьідусьіль іспьів пташок
57 звьідусьіль прийшла уже весна звучить іспьів пташок
58 прийшла уже весна звучидь звьідусьіль іспьів пташок
59 прийшла уже звучидь звьідусьіль весна іспьів пташок
60 прийшла весна звьідусьіль уже іспьів звучить пташок
61 прийшла уже звьідусьіль весна іспьів звучить пташок
62 прийшла весна звучидь звьідусьіль іспьів уже пташок
63 прийшла звучить весна звьідусьіль іспьів уже пташок
64 прийшла звучидь звьідусьіль іспьів уже весна пташок
65 звучить весна звьідусьіль уже іспьів прийшла пташок
66 звучить уже звьідусьіль весна іспьів прийшла пташок
67 звучить уже звьідусьіль іспьів весна прийшла пташок
68 весна уже звьідусьіль звучить іспьів прийшла пташок
69 звьідусьіль уже звучить іспьів весна прийшла пташок
70 весна іспьів уже звучидь звьідусьіль прийшла пташок
71 звучить весна уже пташоґ звьідусьіль іспьів прийшла
72 звучить уже весна пташоґ звьідусьіль іспьів прийшла
73 звучить уже пташок весна іспьів звьідусьіль прийшла
74 звьідусьіль уже звучить весна пташок іспьів прийшла
75 пташоґ звучить весна звьідусьіль уже іспьів прийшла
76 пташоґ звучидь звьідусьіль уже іспьів весна прийшла
77 пташоґ звучить уже звьідусьіль іспьів весна прийшла
78 звьідусьіль пташок уже весна іспьів звучить прийшла
79 пташоґ звучить весна звьідусьіль іспьів уже прийшла
80 пташоґ звучидь звьідусьіль іспьів уже весна прийшла
81 звучить весна звьідусьіль уже іспьів пташок прийшла

82 звучить уже звідусьіль весна іспів пташок прийшла
83 звучить уже звідусьіль іспів весна пташок прийшла
84 весна уже звідусьіль звучить іспів пташок прийшла
85 звідусьіль уже звучить іспів весна пташок прийшла
86 весна іспів уже звучидь звідусьіль пташок прийшла
87 пташоґ звучидь звідусьіль весна уже прийшла іспів
88 пташоґ звучить весна звідусьіль уже прийшла іспів
89 пташоґ звідусьіль звучить уже прийшла весна іспів
90 пташоґ звідусьіль весна уже прийшла звучить іспів
91 пташоґ звідусьіль звучить весна прийшла уже іспів
92 пташоґ звучить весна прийшла звідусьіль уже іспів
93 звідусьіль звучить весна прийшла уже пташок іспів
94 прийшла весна звідусьіль звучить уже пташок іспів
95 прийшла весна звучидь звідусьіль уже пташок іспів
96 прийшла звучидь звідусьіль весна уже пташок іспів
97 прийшла звучить весна звідусьіль уже пташок іспів
98 прийшла уже звідусьіль звучить весна пташок іспів
99 прийшла звідусьіль весна уже пташоґ звучить іспів
100 прийшла весна звучить пташоґ звідусьіль уже іспів
101 прийшла звучить весна пташоґ звідусьіль уже іспів
102 прийшла звучить пташоґ звідусьіль весна уже іспів
103 прийшла весна пташоґ звучить уже звідусьіль іспів
104 прийшла пташоґ звучидь звідусьіль уже весна іспів
105 прийшла весна звідусьіль пташок уже звучить іспів
106 прийшла пташок весна звідусьіль уже звучить іспів

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

16

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут

закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

567

```
1   уже звідусьіль весна прийшла іспівпташоґзвучить
2   прийшла весна звідусьіль уже іспівпташоґзвучить
3   прийшла уже звідусьіль весна іспівпташоґзвучить
```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

012

```
1   прийшлавесна звідусьіль уже іспівпташоґзвучить
```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

прийшлавесна звідусьіль уже іспівпташоґзвучить

```
1   прийшла лунаїе пташок іспів уже весна звідусьіль
2   лунаїе прийшла уже пташок іспів весна звідусьіль
3   прийшла лунаїе пташок уже іспів весна звідусьіль
4   прийшла пташок уже лунаїе іспів весна звідусьіль
5   прийшла лунаїе весна іспів уже пташоґ звідусьіль
6   пташок іспів прийшла весна лунаїе уже звідусьіль
7   лунаїе прийшла весна пташок іспів уже звідусьіль
```

8 прийшла лунаїе пташок весна іспьів уже звьідусьіль
9 прийшла звьідусьіль пташок іспьів лунаїе уже весна
10 пташоґ звьідусьіль прийшла іспьів лунаїе уже весна
11 пташок прийшла іспьів лунаїе уже звьідусьіль весна
12 звьідусьіль прийшла лунаїе пташок іспьів уже весна
13 звьідусьіль лунаїе прийшла пташок іспьів уже весна
14 звьідусьіль пташок лунаїе прийшла іспьів уже весна
15 прийшла пташоґ звьідусьіль уже іспьів лунаїе весна
16 прийшла пташоґ звьідусьіль іспьів уже лунаїе весна
17 лунаїе пташок уже звьідусьіль іспьів прийшла весна
18 звьідусьіль лунаїе пташок іспьів уже прийшла весна
19 пташоґ звьідусьіль лунаїе уже іспьів прийшла весна
20 пташоґ звьідусьіль лунаїе іспьів уже прийшла весна
21 лунаїе прийшла уже звьідусьіль іспьів пташок весна
22 звьідусьіль лунаїе прийшла іспьів уже пташок весна
23 прийшла уже звьідусьіль лунаїе іспьів пташок весна
24 прийшла звьідусьіль лунаїе уже іспьів пташок весна
25 прийшла звьідусьіль лунаїе іспьів уже пташок весна
26 прийшла уже звьідусьіль лунаїе пташок іспьів весна
27 прийшла звьідусьіль лунаїе уже пташок іспьів весна
28 прийшла звьідусьіль лунаїе пташок уже іспьів весна
29 прийшла уже пташоґ звьідусьіль лунаїе іспьів весна
30 пташоґ звьідусьіль лунаїе прийшла уже іспьів весна
31 уже звьідусьіль весна прийшла іспьів пташок лунаїе
32 прийшла весна звьідусьіль уже іспьів пташок лунаїе
33 прийшла уже звьідусьіль весна іспьів пташок лунаїе
34 прийшла уже звьідусьіль іспьів весна пташок лунаїе
35 весна іспьів звьідусьіль прийшла уже пташок лунаїе
36 пташок іспьів прийшла весна звьідусьіль уже лунаїе

37 пташоґ звы́дусьіль прийшла іспы́в весна уже луна́йе
38 прийшла звы́дусьіль пташо́к іспы́в весна уже луна́йе
39 прийшла пташоґ звы́дусьіль іспы́в уже весна луна́йе
40 весна прийшла пташо́к уже іспы́в звы́дусьіль луна́йе
41 прийшла пташоґ звы́дусьіль уже іспы́в весна луна́йе
42 прийшла весна уже пташоґ звы́дусьіль іспы́в луна́йе
43 прийшла уже весна пташоґ звы́дусьіль іспы́в луна́йе
44 прийшла уже пташоґ звы́дусьіль весна іспы́в луна́йе
45 звы́дусьіль пташо́к уже весна іспы́в прийшла луна́йе
46 весна іспы́в звы́дусьіль пташо́к уже прийшла луна́йе
47 звы́дусьіль луна́йе пташо́к іспы́в весна прийшла уже
48 пташоґ звы́дусьіль луна́йе іспы́в весна прийшла уже
49 звы́дусьіль луна́йе прийшла іспы́в весна пташо́к уже
50 прийшла весна звы́дусьіль луна́йе іспы́в пташо́к уже
51 прийшла звы́дусьіль луна́йе іспы́в весна пташо́к уже
52 луна́йе прийшла весна пташоґ звы́дусьіль іспы́в уже
53 луна́йе прийшла весна пташо́к іспы́в звы́дусьіль уже
54 звы́дусьіль луна́йе прийшла пташо́к іспы́в весна уже
55 прийшла весна звы́дусьіль луна́йе пташо́к іспы́в уже
56 прийшла луна́йе пташо́к весна іспы́в звы́дусьіль уже
57 звы́дусьіль прийшла луна́йе пташо́к іспы́в весна уже
58 прийшла луна́йе пташо́к іспы́в весна звы́дусьіль уже
59 прийшла весна пташоґ звы́дусьіль луна́йе іспы́в уже
60 прийшла пташоґ звы́дусьіль весна іспы́в луна́йе уже
61 прийшла пташоґ звы́дусьіль іспы́в весна луна́йе уже
62 прийшла звы́дусьіль пташо́к іспы́в луна́йе весна уже
63 луна́йе пташо́к весна прийшла звы́дусьіль іспы́в уже
64 звы́дусьіль пташо́к луна́йе прийшла іспы́в весна уже
65 пташоґ звы́дусьіль прийшла іспы́в луна́йе весна уже

66 пташок прийшла іспів лунаїе весна звідусьіль уже
67 весна звідусьіль уже прийшла іспів лунаїе пташок
68 звідусьіль лунаїе прийшла весна уже іспів пташок
69 лунаїе прийшла уже звідусьіль іспів весна пташок
70 прийшла звідусьіль лунаїе уже весна іспів пташок
71 прийшла звідусьіль лунаїе уже іспів весна пташок
72 прийшла весна уже звідусьіль лунаїе іспів пташок
73 прийшла уже звідусьіль лунаїе іспів весна пташок
74 прийшла весна звідусьіль уже іспів лунаїе пташок
75 прийшла уже звідусьіль весна іспів лунаїе пташок
76 прийшла уже весна звідусьіль іспів лунаїе пташок
77 звідусьіль лунаїе прийшла іспів весна уже пташок
78 звідусьіль лунаїе прийшла іспів уже весна пташок
79 прийшла весна звідусьіль лунаїе іспів уже пташок
80 прийшла лунаїе весна іспів уже звідусьіль пташок
81 прийшла звідусьіль лунаїе іспів весна уже пташок
82 прийшла звідусьіль лунаїе іспів уже весна пташок
83 прийшла весна звідусьіль іспів уже лунаїе пташок
84 звідусьіль весна лунаїе уже іспів прийшла пташок
85 лунаїе звідусьіль весна уже іспів прийшла пташок
86 лунаїе весна звідусьіль уже іспів прийшла пташок
87 лунаїе уже весна іспів звідусьіль прийшла пташок
88 звідусьіль весна уже лунаїе іспів прийшла пташок
89 звідусьіль уже лунаїе весна іспів прийшла пташок
90 уже лунаїе звідусьіль весна іспів прийшла пташок
91 звідусьіль уже лунаїе іспів весна прийшла пташок
92 звідусьіль весна уже іспів лунаїе прийшла пташок
93 весна уже іспів лунаїе звідусьіль прийшла пташок
94 уже іспів лунаїе звідусьіль весна прийшла пташок

95 звѣдусьіль весна лунаѣе іспѣв уже прийшла пташок
96 іспѣв лунаѣе звѣдусьіль весна уже прийшла пташок
97 звѣдусьіль іспѣв уже лунаѣе весна прийшла пташок
98 звѣдусьіль весна лунаѣе уже пташок іспѣв прийшла
99 весна звѣдусьіль уже пташок іспѣв лунаѣе прийшла
100 лунаѣе пташок уже звѣдусьіль іспѣв весна прийшла
101 пташоґ звѣдусьіль лунаѣе уже весна іспѣв прийшла
102 пташоґ звѣдусьіль лунаѣе уже іспѣв весна прийшла
103 звѣдусьіль лунаѣе пташок іспѣв весна уже прийшла
104 звѣдусьіль лунаѣе пташок іспѣв уже весна прийшла
105 пташоґ звѣдусьіль лунаѣе іспѣв весна уже прийшла
106 пташоґ звѣдусьіль лунаѣе іспѣв уже весна прийшла
107 звѣдусьіль весна лунаѣе уже іспѣв пташок прийшла
108 лунаѣе звѣдусьіль весна уже іспѣв пташок прийшла
109 лунаѣе весна звѣдусьіль уже іспѣв пташок прийшла
110 лунаѣе уже весна іспѣв звѣдусьіль пташок прийшла
111 звѣдусьіль весна уже лунаѣе іспѣв пташок прийшла
112 звѣдусьіль уже лунаѣе весна іспѣв пташок прийшла
113 уже лунаѣе звѣдусьіль весна іспѣв пташок прийшла
114 звѣдусьіль уже лунаѣе іспѣв весна пташок прийшла
115 звѣдусьіль весна уже іспѣв лунаѣе пташок прийшла
116 весна уже іспѣв лунаѣе звѣдусьіль пташок прийшла
117 уже іспѣв лунаѣе звѣдусьіль весна пташок прийшла
118 звѣдусьіль весна лунаѣе іспѣв уже пташок прийшла
119 іспѣв лунаѣе звѣдусьіль весна уже пташок прийшла
120 звѣдусьіль іспѣв уже лунаѣе весна пташок прийшла
121 звѣдусьіль весна лунаѣе уже пташок прийшла іспѣв
122 уже весна пташок лунаѣе прийшла звѣдусьіль іспѣв
123 пташоґ звѣдусьіль лунаѣе уже весна прийшла іспѣв

124 лунаїе пташок весна прийшла звїдусьїль уже їспїв
 125 пташоґ звїдусьїль лунаїе прийшла уже весна їспїв
 126 весна пташок прийшла уже лунаїе звїдусьїль їспїв
 127 звїдусьїль лунаїе прийшла весна уже пташок їспїв
 128 прийшла звїдусьїль лунаїе уже весна пташок їспїв
 129 прийшла звїдусьїль лунаїе уже пташок весна їспїв
 130 прийшла весна уже звїдусьїль лунаїе пташок їспїв
 131 прийшла уже звїдусьїль лунаїе пташок весна їспїв
 132 прийшла весна уже пташоґ звїдусьїль лунаїе їспїв
 133 прийшла уже весна пташоґ звїдусьїль лунаїе їспїв
 134 прийшла уже пташоґ звїдусьїль лунаїе весна їспїв
 135 лунаїе прийшла весна пташоґ звїдусьїль уже їспїв
 136 прийшла весна звїдусьїль лунаїе пташок уже їспїв
 137 прийшла звїдусьїль лунаїе пташок уже весна їспїв
 138 прийшла весна пташоґ звїдусьїль лунаїе уже їспїв
 139 прийшла пташок весна звїдусьїль уже лунаїе їспїв
 140 прийшла пташок уже звїдусьїль весна лунаїе їспїв

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

7

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

5461723

1 їспївпташокужелунаїезвїдусьїльприйшлавесна

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут

закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

1 -10

1 іспівпташокужелунайезвйдусьільприйшлавесна

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

1/-10

?: іспівпташокужелунайезвйдусьільпр ийшлавесна

1 2

?: іспівпташокужелунайезвйдусьіль прийшлавесна

(/ Ви вказали властивості(ість) та діапазон(и) для них такі, що для даних слів та їх сполучень варіант немає. Спробуйте змінити параметри виклику програми (бібліотеки) /)

1

2 лунаїе весна вжеприйшла іспів пташоґ звйдусьіль

3 прийшлавесна звйдусьіль уже іспівпташоґзвучить

4

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

лунаїе весна вжеприйшла іспів пташоґ звйдусьіль

Also here it would be possible to take іспівпташокужелунаєзвідусільприйшлавесна.

So, we have a choice:

1. іспівпташокужелунаєзвідусільприйшлавесна — І спів пташок уже лунає звідусіль: "Прийшла весна!"
2. лунає весна вже прийшла іспів пташок звідусіль — Лунає: "Весна вже прийшла!", — і спів пташок звідусіль
3. прийшлавесна звідусіль уже іспівпташокзвучить — Прийшла весна, звідусіль уже і спів пташок звучить

Роем "Happy Tree" ("Щасливе дерево")

Another example.

Consider the poem by Oleksandr (Serhiyovych) Zhabenko "Happy Tree".

— Яке воно — щасливе дерево?
 — Ти можеш бачити — цвіте воно;
 Закриєш очі — плодоносить.
 Радітимеш, побачивши його
 і квітним, і водночас плідним.

— А навесні, коли ще квіту там нема,
 Чи є щасливим дерево?
 — Напевне. У надії, в тій порі,
 що сонце гріє. І вранці
 сяють теплі й прохолодні роси.

Приходиш ти до нього навесні
 і бачиш дерево щасливим.
 Прийдеш улітку, восени
 і знову зустрічаєш його милим,

усміхненим, красивим і
дзвінким від співу птахів.

Уранці, ввечері чи вдень
– постійно бачиш його квітним.
Підійдеш ближче – ось воно!
Відходиш далі – хочеш бути знов.

– А взимку чи щасливе дерево?
– Так, бо земля його тримає,
А сніг тепло оберігає та радує людей,
свята Різдва радість огортає.
Для дерева затих той теплий рух,
Та сили додає йому той зріст,
і гнучкість, і міцне здоров'я.

Це пори року й дерево просте. А ми – люди:
Багатші, і рухливі, і повсюди.
У щасті ми активні,
а як ні, – йдемо до нього
поступово,
людяно.

(2021 р.)

```
propertiesTextG3 Haskell/old/Shchaslyve_derevo.txt s 1 0 +d 56.csv HK02 | distributionTextG s 1 +W
5
32080.0000 281681185542160.0000 40037543026991875000000000.0000 8780585584.2319 1248053086876305500000
0 142137796494.7515 0.00000000 4 1 Яке воно щасливе дерево
8080.0000 4012012044040.0000 39087701482596790000000000000000.0000 496536144.0644 4837586817153068000
```

000000.0000 97426680312844750000.0000 0.00000000 5 1 Ти можеш бачити цвіте воно
 32080.0000 40046154607286250000000000.0000 40046154607286250000000000.0000 1248321527658548800000.0000
 124
 8321527658548800000.0000 1.0000 2.00000000 3 5 Закриєш очі плодоносить
 80080.0000 3302678097230517000000000000000000.0000 3302678097230517000000000000000000.0000 412422339814000
 000000000000.0000 412422339814000660000000000000.0000 1.0000 2.00000000 3 5 Радітимеш побачивши
 його
 2986128.0000 64192192104040.0000 3908827887225641000000000000000000.0000 21496798.5646 130899542391539830
 0000000.0000 6089257523548000000.0000 0.00000000 3 1 іквітним іводночас плідним
 0.0000 0.0000 5403583893436425000000000000000000000000000000.0000 NaN Infinity Infinity 0.00000000 4 1 Анавесні
 колищеквіту там нема
 8080.0000 10859489273691456000000.0000 3302678097230517000000000000000000.0000 1343996197239041500.0000
 408
 7472892612026000000000000000.0000 30412830787.8317 0.00000000 3 1 Чийе щасливим дерево
 128080.0000 746640.0000 16819198377986623000.0000 5.8295 131317913632000.5000 22526516631826.0800 0.0000
 00 4 1 Напевне Унадії втій порі
 32080.0000 32080.0000 66193821698576040.0000 1.0000 2063398432000.5000 2063398432000.5000 0.00000000
 3 1
 щосонце гріє Івранці
 32080.0000 8274227712336040.0000 3908770148259679000000000000000000.0000 257924804000.5000 12184445599313
 500000000000.0000 47240301864451910.0000 0.00000000 4 1 сяють теплі йпрохолодні роси
 8080.0000 80080.0000 3908827887225641000000000000000000.0000 9.9109 483765827626935800000000000000.0000
 4881
 15370532672400000000000000.0000 0.00000000 4 1 Приходиш ти донього навесні
 8080.0000 64192192104040.0000 3302678097230517000000000000000000.0000 7944578230.6980 4087472892612026000
 00000000.0000 5144984131212836000.0000 0.00000000 3 1 ібачиш дерево щасливим
 8080.0000 128080.0000 195497447640089100000000.0000 15.8515 2419522866832786000.0000 152636982854535550.0
 0 0.00000000 3 1 Прийдеш улітку восени
 186768.0000 16778002444290623000.0000 3908827887225641000000000000000000.0000 89833389254533.0200 2092878

23847749931.6604 0.00000000 3 1 ігнучкість імічне здоров'я
0.0000 0.0000 Infinity NaN Infinity Infinity 0.00000000 6 1 Цепори року йдереве просте Ами люд
и
281217040.0000 1124865040.0000 27799367259008490000000000.0000 4.0000 9885377948295200.0000 2471351341758
2.5000 0.00000000 3 1 Багатші ірухливі іповсюди
5473632528.0000 12812904023400.0000 43437957094765720000000.0000 2340.8411 7935855553430.3340 3390172674
039 0.00000000 3 1 Ущасті ми активні
16048048056040.0000 16048048056040.0000 266394626564040.0000 1.0000 16.5998 16.5998 0.11363756 2 1 айакн
йдемо донього
поступово
людяно

1 2 3 4 5
29 0 0 0 2
93.55% 0.00% 0.00% 0.00% 6.45%
0.1327+-0.4989 0 31
2 3 4 5 6 7
1 14 10 5 1 0

1
12 . . . 2
10
5
1
. . . .

~~~~~  
1 0 0 0 0  
12 0 0 0 2  
10 0 0 0 0

```

5 0 0 0 0
1 0 0 0 0
0 0 0 0 0

```

---

As you can see, the presence of Infinity often allows you to get only limited information. Let's look, for example, at the following line: "Для дерева затих той теплый рух".

```

lineVariantsG3 +d 56.csv 3 НК02 Для дерева затих той теплый рух
0
затих теплый той Длядерева рух
3302678097230517000000000000000000.0000
3302678097230517000000000000000000.0000
затих той теплый Длядерева рух
40037543026991875000000000.0000
40037543026991875000000000.0000
рух затих теплый той Длядерева
40037543026991875000000000.0000
40037543026991875000000000.0000
рух затих теплый Длядерева той
40037543026991875000000000.0000
40037543026991875000000000.0000
рух теплый затих той Длядерева
27799367259008490000000000.0000
27799367259008490000000000.0000

```

It is inconvenient to work with large numbers, and with Infinity in the approach of intervals it is even more inconvenient. Therefore, we will use another additional property.

lineVariantsG3 +d 56.csv +m GHK02 2118 НК02 15 -m Для дерева зати́х той теплий рух  
0

зати́х рух теплий дльа́дерева той  
зати́х теплий той дльа́дерева рух  
зати́х той теплий дльа́дерева рух  
руг зати́х теплий дльа́дерева той  
руг зати́х той дльа́дерева теплий  
рух теплий той дльа́дерева зати́х  
зати́х рух теплий той дльа́дерева  
руг зати́х теплий той дльа́дерева  
руг зати́х той теплий дльа́дерева  
рух теплий зати́х той дльа́дерева

or even the following variant:

lineVariantsG3 +d 56.csv +m GHK02 2118 НК02 15 GHK02 23 -m Для дерева зати́х той теплий рух  
0

зати́х той теплий дльа́дерева рух  
руг зати́х теплий дльа́дерева той  
руг зати́х той дльа́дерева теплий  
руг зати́х теплий той дльа́дерева  
рух теплий зати́х той дльа́дерева

You can see that among these several options interesting in sound is the "рух зати́х той Для́дерева теплий", but the replacement does not fit into the overall rhythmic pattern of the poem, so it can not be recommended here.

## General Structure of the Additional Properties

We describe the general structure of additional properties.



It is also defined by alphanumeric and numeric arguments, but always consists of only two components.

The first – the designation of the type of additional property – to the type of property is added in writing among the letters (usually convenient to start with) the Latin letter  $\acute{G}$ . For example, to apply an additional property to the property Hw02, denote it GHw02.

The second – anti-interval string sampling (new term) – a sequence of decimal digits, which is understood by the program lineVariantsG3 according to the table:

| The first digit | Inequality type | The second digit | Expression                                | Parameter 1 | Parameter 2 |
|-----------------|-----------------|------------------|-------------------------------------------|-------------|-------------|
| 0               | >               |                  |                                           |             |             |
| 1               | <               |                  |                                           |             |             |
| 2               | >               | 1                | $\lg y >$                                 |             |             |
|                 |                 | 2                | $637.0 \cdot y > base$                    | $base$      |             |
|                 |                 | 3                | $100.0 \cdot \sin(ky) > 0.5 \cdot base1$  | $k$         | $base1$     |
|                 |                 | 4                | $100.0 \cdot \cos(ky) > 0.5 \cdot base1$  | $k$         | $base1$     |
|                 |                 | 5                | $1000.0 \cdot \sin(ky) > base2$           | $k$         | $base2$     |
|                 |                 | 6                | $1000.0 \cdot \cos(ky) > base2$           | $k$         | $base2$     |
|                 |                 | 7                | $100.0 \cdot \sin(ky) > -0.5 \cdot base1$ | $k$         | $base1$     |
|                 |                 | 8                | $100.0 \cdot \cos(ky) > -0.5 \cdot base1$ | $k$         | $base1$     |
|                 |                 | –                | $y^k > base1$                             | $k$         | $base1$     |
| –               | <               | 1                | $\lg y <$                                 |             |             |
|                 |                 | 2                | $637.0 \cdot y < base$                    | $base$      |             |
|                 |                 | 3                | $100.0 \cdot \sin(ky) < 0.5 \cdot base1$  | $k$         | $base1$     |
|                 |                 | 4                | $100.0 \cdot \cos(ky) < 0.5 \cdot base1$  | $k$         | $base1$     |
|                 |                 | 5                | $1000.0 \cdot \sin(ky) < base2$           | $k$         | $base2$     |
|                 |                 | 6                | $1000.0 \cdot \cos(ky) < base2$           | $k$         | $base2$     |
|                 |                 | 7                | $100.0 \cdot \sin(ky) < -0.5 \cdot base1$ | $k$         | $base1$     |
|                 |                 | 8                | $100.0 \cdot \cos(ky) < -0.5 \cdot base1$ | $k$         | $base1$     |
|                 |                 | –                | $y^k < base1$                             | $k$         | $base1$     |

where  $base > 0, base \in N$ ,  
 $base1 \in [1..100], base1 \in N$ ,  
 $base2 \in [1..1000], base2 \in N$ ,  
 $k \in [0..9]$  — a digit,  $base1 > 0, base1 \in N$ .

Underline in the table means any number other than those above in the same column.

Thus, for example, GHW02 2118 means that the Hw02 property will be used for calculation, but only those values that satisfy the inequality without permutation of intervals will be taken:  $\lg y > 18$ , ie numbers whose order is greater than 18 (they greater than 1,000,000,000,000,000,000) (including Infinity). Gt32 01 means that for the property t32 will be taken only those whose values satisfy the inequality  $y > 1$ . Gw02 0100000 and Gw02 215 are actually equivalent, but the former does not calculate the decimal logarithm (it can be expected to run slightly faster).

## Syllable-as-a-whole (SaaW) mode

is mode is called by setting among the properties at least one with an additional letter 'a' in the designation and an additional setting as a command line argument consecutively outside the groups +m ...-m and +a ...-a of the arguments "+s" a natural number greater 0 is the number of sets of groups of "syllable-number" pairs.

Allows you to set both syllable durations and prosodic properties, for example, logical stresses, dynamic stresses, melodic stresses (although the latter method is still in the process of research), intonation features, general pronunciation characteristics, etc. At the same time, in many cases, the system of numbers "2-3-4-6-8-11-15" is convenient, each of which is greater than the previous one in no less than 1.3 times. If you skip entering a number, it is accepted by the program as 4. You can also specify floating-point numbers, the format is Double (but given the way most properties are calculated, this will most likely be excessive precision, and therefore you can limit yourself to only these numbers).

For example, if you set the strength of emphasizing of syllables, where the syllable next to the accented one in a logically unstressed word corresponds to the default value of 4.0 (just press Enter in the place of an empty number, which is then accepted by the program as 4; 6 - an accented syllable in a logically unaccented word, 8 - a stressed syllable in a logically stressed word with a slight amplification, 3 - the syllable is pronounced a little "casually", it is distant from the stressed one):

```
lineVariantsG3 3 +i a01y +s 1 десь тут була подоляночка
0
```

? десь 6  
 ? туд  
 ? бу  
 ? ла 6  
 ? по 3  
 ? до  
 ? лья 8  
 ? но 6  
 ? чка  
 1 подольаночка дезь була тут  
 2 десь туд була подольаночка  
 3 туд була десь подольаночка  
 4 дезь була тут подольаночка  
 5 десь подольаночка туд була  
 6 тут подольаночка дезь була

Please, specify the variant which you would like to become the resulting string by its number.

2  
 десь туд була подольаночка

Please note that the words are written "in phonetic transcription", i.e. not according to the rules of orthography, and by the peculiarities of pronunciation.

**N.B.:** This is a good way to see how the text actually sounds in Ukrainian (with certain remarks, which will be discussed later in the text).

lineVariantsG3 0.001\_100.0 3 +i a01y +s 1 десь тут була подольаночка  
 0  
 ? десь 6  
 ? туд  
 ? бу

? ла 6  
 ? по 3  
 ? до  
 ? лья 8  
 ? но 6  
 ? чка

1 подольаночка дезь була тут  
 2 дезь туд була подольаночка  
 3 туд була дезь подольаночка  
 4 дезь була тут подольаночка  
 5 дезь подольаночка туд була  
 6 тут подольаночка дезь була

Please, specify the variant which you would like to become the resulting string by its number.

2

дезь туд була подольаночка

It is interesting that from the point of view of this property, the accents are more rhythmic according to the three-syllabic pattern, that is, they correspond more to the three-syllabic meter (in this case, the dactyl).

Another example of a similar approach:

lineVariantsG3 3 aw01 +s 1 садок вишневий колохати

0

? са  
 ? док 6  
 ? ви  
 ? шне 6  
 ? вий  
 ? ко 6  
 ? ло

```

? ха 6
? ти
 садок вишневий колохати
4.0000
4.0000
 вишневий колохати садок
0.8125
0.8125
 вишневий садок колохати
0.4375
0.4375

```

Let's consider in more detail the value of the parameters with which the mode is called.

The number after "+s" must be a natural number, not less than 1 — it indicates the number of sets of "syllable-number" pairs that encode the calculation. If you specify 1, as in the examples above, then such a set is only one - therefore, during execution, you will be prompted to enter numerical values for syllables only once in a row. If you say 2, then it will be offered twice, 3 - three times, etc. At the same time, the last number in the designation of properties with the letter 'a' must not be greater than this number. It does not make sense to specify a number greater than the largest last digit in the record of such properties, because it will lead to redundant data entry that will not be used anywhere, which will simply take time.

Another example, in this case with 2:

```

lineVariantsG3 100.0_-100.0 3 a02y +s 2 хруці над вишнями гудуть
0
? хру
? шчьі 6
? на
? дви 6
? шньа
? ми 6
? гу

```

? дуть 6  
 ? хру  
 ? шчыі 6  
 ? на  
 ? дви 6  
 ? шньа  
 ? ми 6  
 ? гу 6  
 ? дуть 8  
 гудуть хруці надвишнями  
 924.0000  
 924.0000  
 хруці гудуть надвишнями  
 924.0000  
 924.0000  
 хруці надвишнями гудуть  
 624.0000  
 624.0000  
 надвишнями гудуть хруці  
 624.0000  
 624.0000  
 надвишнями хруці гудуть  
 -355.5556  
 -355.5556  
 гудуть надвишнями хруці  
 -355.5556  
 -355.5556

If we are to understand that the two-syllables meter is expected here, then it is reasonable to call the program with somewhat like 10.0\_0.0 first argument to eliminate the influence of the three-syllable-ness.

lineVariantsG3 100.0\_0.0 3 a02y +s 2 хруці над вишнями гудуть

0

? хру

? шчыі 6

? на

? дви 6

? шньа

? ми 3

? гу

? дуть 6

? хру

? шчыі 6

? на

? дви 6

? шньа

? ми 6

? гу

? дуть 6

хруці надвишнями гудуть

1024.0000

1024.0000

надвишнями гудуть хруці

1024.0000

1024.0000

гудуть хруці надвишнями

1024.0000

1024.0000

хруці гудуть надвишнями

1024.0000

1024.0000

надвишнями хрущі гудуть

44.4444

44.4444

гудуть надвишнями хрущі

44.4444

44.4444

Moreover, here we see that in the maximum position (the top-most options) there are the most of the options possible, that in general case makes the usage not so desirable.

Please note that it was suggested to enter data twice for the same syllables, and data from the second input was used.

It seems that Taras Shevchenko repeats the semantic structure of the first line to create an idyllic harmonic mood. Correspondence between: хрущі – садок, вишневий – над вишнями, (є) колохати – гудуть emphasises the thought in his authentic option.

On the other hand, the combined property, taking into account both two-syllable and three-syllable meter, gives the following:

lineVariantsG3 3 a02y +s 2 хрущі над вишнями гудуть

0

? хру

? шчыі 6

? на

? дви 6

? шнья

? ми 3

? гу

? дуть 6

? хру

? шчыі 6

? на

? дви 6

? шнья



? ми 3  
 ? гу 6  
 ? дуть 8  
 хрущі надвишнями гудуть  
 14.2400  
 14.2400  
 надвишнями гудуть хрущі  
 14.2400  
 14.2400  
 гудуть хрущі надвишнями  
 11.2400  
 11.2400  
 хрущі гудуть надвишнями  
 11.2400  
 11.2400  
 надвишнями хрущі гудуть  
 4.4444  
 4.4444  
 гудуть надвишнями хрущі  
 4.4444  
 4.4444

## SaaW in multiple properties mode

All of the above also applies to the multiple properties mode. At the same time, the letter 'H' in the designation of the property, if 'a' is used in it, is ignored, that is, SaaW has priority over the PhoPaaW ('H') own durations mode.

Example:

```
lineVariantsG3 100.0_0.001 +m a01x 5 02y 20 -m +s 1 +v 3 якісь цікаві ми отримуємо результати
3
```

? йа  
 ? кьісь 6  
 ? цьі  
 ? ка 6  
 ? вьі  
 ? ми 6  
 ? о  
 ? три 6  
 ? му  
 ? йе 5  
 ? мо  
 ? ре  
 ? зуль 6  
 ? та 8  
 ? ти

ми цікаві отримуємо якісь результати ->144.25->144.25  
 отримуємо якісь цікаві ми результати ->144.25->144.25  
 ми результати якісь цікаві отримуємо ->144.25->144.25  
 якісь отримуємо цікаві ми результати ->144.02777777777777->144.02777777777777  
 результати цікаві ми якісь отримуємо ->144.02777777777777->144.02777777777777  
 якісь результати ми цікаві отримуємо ->144.02777777777777->144.02777777777777  
 результати ми якісь цікаві отримуємо ->144.02777777777777->144.02777777777777  
 ми цікаві результати якісь отримуємо ->144.02777777777777->144.02777777777777  
 цікаві якісь отримуємо ми результати ->82.36111111111111->82.36111111111111  
 ми результати отримуємо цікаві якісь ->82.36111111111111->82.36111111111111  
 ми отримуємо цікаві результати якісь ->82.36111111111111->82.36111111111111  
 цікаві результати отримуємо ми якісь ->82.36111111111111->82.36111111111111  
 ми результати якісь отримуємо цікаві ->82.36111111111111->82.36111111111111  
 ми якісь результати отримуємо цікаві ->82.36111111111111->82.36111111111111

якісь цікаві отримуємо ми результати ->81.69444444444444->81.69444444444444  
цікаві отримуємо якісь ми результати ->81.69444444444444->81.69444444444444  
отримуємо якісь ми цікаві результати ->81.69444444444444->81.69444444444444  
якісь ми отримуємо цікаві результати ->81.69444444444444->81.69444444444444  
отримуємо результати ми цікаві якісь ->81.69444444444444->81.69444444444444  
результати ми отримуємо цікаві якісь ->81.69444444444444->81.69444444444444  
цікаві ми отримуємо результати якісь ->81.69444444444444->81.69444444444444  
результати цікаві отримуємо ми якісь ->81.69444444444444->81.69444444444444  
цікаві результати якісь ми отримуємо ->81.69444444444444->81.69444444444444  
отримуємо якісь результати ми цікаві ->81.69444444444444->81.69444444444444  
результати ми якісь отримуємо цікаві ->81.69444444444444->81.69444444444444  
ми результати отримуємо якісь цікаві ->81.69444444444444->81.69444444444444  
якісь цікаві результати отримуємо ми ->81.69444444444444->81.69444444444444  
якісь цікаві ми отримуємо результати ->81.25->81.25  
отримуємо цікаві ми якісь результати ->81.25->81.25  
якісь отримуємо ми цікаві результати ->81.25->81.25  
отримуємо ми якісь цікаві результати ->81.25->81.25  
результати цікаві ми отримуємо якісь ->81.25->81.25  
цікаві ми результати отримуємо якісь ->81.25->81.25  
ми цікаві отримуємо результати якісь ->81.25->81.25  
отримуємо цікаві ми результати якісь ->81.25->81.25  
цікаві результати ми якісь отримуємо ->81.25->81.25  
ми якісь результати цікаві отримуємо ->81.25->81.25  
отримуємо якісь ми результати цікаві ->81.25->81.25  
результати якісь ми отримуємо цікаві ->81.25->81.25  
ми отримуємо результати якісь цікаві ->81.25->81.25  
отримуємо результати якісь цікаві ми ->81.25->81.25  
отримуємо якісь результати цікаві ми ->81.25->81.25  
результати цікаві отримуємо якісь ми ->81.25->81.25

цікаві отримуємо якісь результати ми ->81.25->81.25

отримуємо цікаві якісь ми результати ->10000.036->10000.036  
результати ми отримуємо цікаві якісь ->3600.144->3600.144  
отримуємо цікаві якісь результати ми ->3600.144->3600.144  
цікаві результати якісь отримуємо ми ->3600.144->3600.144  
ми отримуємо цікаві результати якісь ->3600.081->3600.081  
ми цікаві результати отримуємо якісь ->3600.081->3600.081  
результати цікаві отримуємо ми якісь ->3600.081->3600.081  
цікаві отримуємо результати ми якісь ->3600.081->3600.081  
отримуємо цікаві ми результати якісь ->3600.081->3600.081  
якісь ми результати цікаві отримуємо ->3600.081->3600.081  
ми результати цікаві якісь отримуємо ->3600.081->3600.081  
ми цікаві результати якісь отримуємо ->3600.081->3600.081  
ми якісь результати отримуємо цікаві ->3600.081->3600.081  
ми якісь отримуємо результати цікаві ->3600.081->3600.081  
результати якісь отримуємо цікаві ми ->3600.081->3600.081  
результати отримуємо якісь цікаві ми ->3600.081->3600.081  
отримуємо якісь результати цікаві ми ->3600.081->3600.081  
отримуємо якісь цікаві результати ми ->3600.081->3600.081  
результати цікаві якісь отримуємо ми ->3600.081->3600.081  
отримуємо цікаві ми якісь результати ->3600.036->3600.036  
ми якісь отримуємо цікаві результати ->3600.036->3600.036  
ми результати отримуємо цікаві якісь ->3600.036->3600.036  
ми отримуємо результати цікаві якісь ->3600.036->3600.036  
результати якісь ми цікаві отримуємо ->3600.036->3600.036  
результати ми цікаві якісь отримуємо ->3600.036->3600.036  
ми цікаві якісь результати отримуємо ->3600.036->3600.036  
результати ми якісь отримуємо цікаві ->3600.036->3600.036

якісь результати отримуємо цікаві ми ->3600.036->3600.036  
якісь отримуємо результати цікаві ми ->3600.036->3600.036  
якісь отримуємо цікаві результати ми ->3600.036->3600.036  
отримуємо результати якісь цікаві ми ->3600.036->3600.036  
якісь цікаві результати отримуємо ми ->3600.036->3600.036  
цікаві якісь результати отримуємо ми ->3600.036->3600.036  
цікаві якісь отримуємо результати ми ->3600.036->3600.036  
якісь отримуємо цікаві ми результати ->625.081->625.081  
цікаві результати отримуємо ми якісь ->625.081->625.081  
ми цікаві якісь отримуємо результати ->625.036->625.036  
якісь ми цікаві отримуємо результати ->625.036->625.036  
отримуємо якісь цікаві ми результати ->625.036->625.036  
отримуємо ми цікаві результати якісь ->400.144->400.144  
цікаві результати ми отримуємо якісь ->400.144->400.144  
цікаві ми якісь отримуємо результати ->400.081->400.081  
отримуємо ми результати цікаві якісь ->400.081->400.081  
результати цікаві ми отримуємо якісь ->400.081->400.081  
цікаві ми отримуємо результати якісь ->400.081->400.081  
результати якісь цікаві ми отримуємо ->400.081->400.081  
якісь результати цікаві ми отримуємо ->400.081->400.081  
якісь цікаві результати ми отримуємо ->400.081->400.081  
якісь цікаві ми результати отримуємо ->400.081->400.081  
цікаві якісь результати ми отримуємо ->400.081->400.081  
цікаві ми результати якісь отримуємо ->400.081->400.081  
результати ми якісь цікаві отримуємо ->400.081->400.081  
ми якісь результати цікаві отримуємо ->400.081->400.081  
отримуємо якісь ми результати цікаві ->400.081->400.081  
результати отримуємо ми якісь цікаві ->400.081->400.081  
отримуємо результати ми якісь цікаві ->400.081->400.081

результати якісь ми отримуємо цікаві ->400.081->400.081  
результати цікаві отримуємо якісь ми ->400.081->400.081  
цікаві якісь ми отримуємо результати ->400.036->400.036  
отримуємо якісь ми цікаві результати ->400.036->400.036  
якісь ми отримуємо цікаві результати ->400.036->400.036  
отримуємо ми якісь цікаві результати ->400.036->400.036  
результати отримуємо ми цікаві якісь ->400.036->400.036  
отримуємо результати ми цікаві якісь ->400.036->400.036  
цікаві ми результати отримуємо якісь ->400.036->400.036  
результати отримуємо якісь ми цікаві ->400.036->400.036  
отримуємо результати якісь ми цікаві ->400.036->400.036  
отримуємо ми якісь результати цікаві ->400.036->400.036  
якісь ми результати отримуємо цікаві ->400.036->400.036  
якісь ми отримуємо результати цікаві ->400.036->400.036  
цікаві отримуємо результати якісь ми ->400.036->400.036  
ми результати якісь цікаві отримуємо ->225.144->225.144  
ми якісь цікаві отримуємо результати ->225.081->225.081  
результати ми цікаві отримуємо якісь ->225.081->225.081  
ми результати цікаві отримуємо якісь ->225.081->225.081  
цікаві результати якісь ми отримуємо ->225.081->225.081  
якісь результати ми цікаві отримуємо ->225.081->225.081  
ми цікаві отримуємо результати якісь ->225.036->225.036  
результати цікаві якісь ми отримуємо ->225.036->225.036  
цікаві якісь ми результати отримуємо ->225.036->225.036  
якісь ми цікаві результати отримуємо ->225.036->225.036  
результати отримуємо цікаві якісь ми ->225.036->225.036  
отримуємо результати цікаві якісь ми ->225.036->225.036  
отримуємо цікаві результати якісь ми ->225.036->225.036  
цікаві результати отримуємо якісь ми ->225.036->225.036

ми цікаві отримуємо якісь результати ->156.331->156.331  
цікаві отримуємо ми результати якісь ->25.144->25.144  
результати якісь отримуємо ми цікаві ->25.144->25.144  
отримуємо ми результати якісь цікаві ->25.144->25.144  
результати ми отримуємо якісь цікаві ->25.144->25.144  
результати якісь цікаві отримуємо ми ->25.144->25.144  
ми отримуємо цікаві якісь результати ->25.081->25.081  
цікаві ми отримуємо якісь результати ->25.081->25.081  
цікаві отримуємо якісь ми результати ->25.081->25.081  
цікаві якісь отримуємо ми результати ->25.081->25.081  
якісь отримуємо ми цікаві результати ->25.081->25.081  
ми отримуємо якісь цікаві результати ->25.081->25.081  
результати отримуємо цікаві ми якісь ->25.081->25.081  
отримуємо результати цікаві ми якісь ->25.081->25.081  
результати цікаві ми якісь отримуємо ->25.081->25.081  
цікаві ми якісь результати отримуємо ->25.081->25.081  
ми якісь цікаві результати отримуємо ->25.081->25.081  
якісь отримуємо ми результати цікаві ->25.081->25.081  
отримуємо якісь результати ми цікаві ->25.081->25.081  
якісь результати ми отримуємо цікаві ->25.081->25.081  
ми результати якісь отримуємо цікаві ->25.081->25.081  
ми отримуємо результати якісь цікаві ->25.081->25.081  
ми отримуємо якісь результати цікаві ->25.081->25.081  
цікаві отримуємо якісь результати ми ->25.081->25.081  
якісь цікаві ми отримуємо результати ->25.036->25.036  
отримуємо ми цікаві якісь результати ->25.036->25.036  
цікаві отримуємо ми якісь результати ->25.036->25.036  
отримуємо цікаві результати ми якісь ->25.036->25.036  
цікаві результати ми якісь отримуємо ->25.036->25.036

якісь результати отримуємо ми цікаві ->25.036->25.036  
якісь отримуємо результати ми цікаві ->25.036->25.036  
ми результати отримуємо якісь цікаві ->25.036->25.036  
якісь результати цікаві отримуємо ми ->25.036->25.036  
якісь цікаві отримуємо результати ми ->25.036->25.036  
якісь цікаві отримуємо ми результати ->6.394->6.394

ми отримуємо результати йакьісь цыікавыі  
ми отримуємо цыікавыі результати йакьісь  
ми результати отримуємо цыікавыі йакьісь  
ми результати отримуємо йакьісь цыікавыі  
ми цыікавыі отримуємо результати йакьісь  
ми цыікавыі отримуємо йакьісь результати  
ми результати йакьісь отримуємо цыікавыі  
ми йакьісь результати отримуємо цыікавыі  
ми результати йакьісь цыікавыі отримуємо  
ми цыікавыі результати йакьісь отримуємо  
ми йакьісь результати цыікавыі отримуємо  
отримуємо ми йакьісь цыікавыі результати  
результати ми отримуємо цыікавыі йакьісь  
цыікавыі ми отримуємо результати йакьісь  
йакьісь ми отримуємо цыікавыі результати  
результати ми йакьісь отримуємо цыікавыі  
цыікавыі ми результати отримуємо йакьісь  
результати ми йакьісь цыікавыі отримуємо  
отримуємо результати ми цыікавыі йакьісь  
отримуємо цыікавыі ми результати йакьісь  
отримуємо йакьісь ми результати цыікавыі  
отримуємо цыікавыі ми йакьісь результати



отримуюємо йакьісь ми цыікавыі результати  
 йакьісь отримуємо ми цыікавыі результати  
 результати цыікавыі ми отримуємо йакьісь  
 результати йакьісь ми отримуємо цыікавыі  
 йакьісь цыікавыі ми отримуємо результати  
 результати цыікавыі ми йакьісь отримуємо  
 цыікавыі результати ми йакьісь отримуємо  
 йакьісь результати ми цыікавыі отримуємо  
 отримуємо йакьісь результати ми цыікавыі  
 отримуємо йакьісь цыікавыі ми результати  
 цыікавыі отримуємо йакьісь ми результати  
 йакьісь отримуємо цыікавыі ми результати  
 результати цыікавыі отримуємо ми йакьісь  
 цыікавыі результати отримуємо ми йакьісь  
 цыікавыі йакьісь отримуємо ми результати  
 йакьісь цыікавыі отримуємо ми результати  
 цыікавыі результати йакьісь ми отримуємо  
 отримуємо результати йакьісь цыікавыі ми  
 отримуємо йакьісь результати цыікавыі ми  
 цыікавыі отримуємо йакьісь результати ми  
 результати цыікавыі отримуємо йакьісь ми  
 йакьісь цыікавыі результати отримуємо ми

It is worth noting that due to the peculiarities of the calculation of properties, if you give the same values to the syllables in a row, then there will be fewer values with maximum values. The question whether it is worth trying too hard to give different values to consecutive syllables remains open.

I will also note that such a detailed output is provided on the condition that it is specified as arguments "+v 3". It includes "detailed" output, which will be discussed later. It works precisely in the mode of multiple properties.

Another interesting example is one that shows how you can try to find where the logical emphasis would be better:

```
lineVariantsG3 10.0_0.0 +m a01x 5 a02x 5 a03x 5 -m +s 3 +v 3 хруці над вишнями гудуть
```

3  
 ? хру 6  
 ? шчыі 8  
 ? на  
 ? дви 6  
 ? шньа  
 ? ми 3  
 ? гу  
 ? дуть 6  
 ? хру  
 ? шчыі 6  
 ? на 6  
 ? дви 8  
 ? шньа 6  
 ? ми  
 ? гу  
 ? дуть 6  
 ? хру  
 ? шчыі 6  
 ? на  
 ? дви 6  
 ? шньа  
 ? ми 3  
 ? гу 6  
 ? дуть 8

хруці надвишнями гудуть ->14.240000000000002->14.240000000000002  
 надвишнями гудуть хруці ->14.240000000000002->14.240000000000002  
 гудуть хруці надвишнями ->11.240000000000002->11.240000000000002  
 хруці гудуть надвишнями ->11.240000000000002->11.240000000000002

надвишнями хрущі гудуть ->4.4444444444444445->4.4444444444444445  
 гудуть надвишнями хрущі ->4.4444444444444445->4.4444444444444445

хрущі надвишнями гудуть ->14.2400000000000002->14.2400000000000002  
 надвишнями гудуть хрущі ->14.2400000000000002->14.2400000000000002  
 гудуть хрущі надвишнями ->11.2400000000000002->11.2400000000000002  
 хрущі гудуть надвишнями ->11.2400000000000002->11.2400000000000002  
 надвишнями хрущі гудуть ->4.4444444444444445->4.4444444444444445  
 гудуть надвишнями хрущі ->4.4444444444444445->4.4444444444444445

хрущі надвишнями гудуть ->14.2400000000000002->14.2400000000000002  
 надвишнями гудуть хрущі ->14.2400000000000002->14.2400000000000002  
 гудуть хрущі надвишнями ->11.2400000000000002->11.2400000000000002  
 хрущі гудуть надвишнями ->11.2400000000000002->11.2400000000000002  
 надвишнями хрущі гудуть ->4.4444444444444445->4.4444444444444445  
 гудуть надвишнями хрущі ->4.4444444444444445->4.4444444444444445

гудуть надвишньами хрушчыі  
 гудуть хрушчыі надвишньами  
 надвишньами гудуть хрушчыі  
 хрушчыі гудуть надвишньами  
 надвишньами хрушчыі гудуть  
 хрушчыі надвишньами гудуть

As you can see, the calculation results for all three cases do not differ, but in general they could be different, which may indicate the greater expediency of one or another logical emphasis (which is set by shifting to the right one value position for the scale "2-3-4-6-8-11-15").

## Recursive mode in SaaW

The recursive mode is characterized by two essential features:

1. since the line is written in "phonetic transcription", the next calculation call will be made on the basis of a new line, in which irreversible phonetic changes may occur (the most difficult is the case of voiceless consonants becoming voiced, characteristic of the Ukrainian language, for other languages there may be other characteristic changes);
2. each subsequent stage of calculations in SaaW mode will cause manual input of data for warehouses – this lengthens the total duration of calculations, increases human work, but at the same time allows changing the data to get modified results.

The first complication may not affect the results of the work very much (because for 2-4 changes in a long line and considering that the differences between voiceless and corresponding voiced consonants are no more than 11%, it is mostly not too strongly reflected in the final results, although and the influence cannot be completely neglected).

Example:

```
lineVariantsG3 +r 3 as01 +s 1 +u малювати щось гарне це цікаве заняття
0
? ма 3
? льу
? ва 6
? ти
? шчозь 6
? гар 6
? не
? це
? цьі
? ка 6
? ве
? за
? нья 6
```

? тьтя

1 гарне щось цецьікаве мальувати заньатъття  
 2 цецьікаве щозь гарне мальувати заньатъття  
 3 щозь гарне цецьікаве мальувати заньатъття  
 4 заньатъття щозь гарне цецьікаве мальувати  
 5 щозь гарне заньатъття цецьікаве мальувати  
 6 щозь гарне цецьікаве заньатъття мальувати  
 7 заньатъття гарне щось цецьікаве мальувати  
 8 гарне заньатъття щось цецьікаве мальувати  
 9 щось цецьікаве гарне заньатъття мальувати  
 10 заньатъття щозь гарне мальувати цецьікаве  
 11 щозь гарне заньатъття мальувати цецьікаве  
 12 заньатъття гарне щось мальувати цецьікаве  
 13 гарне заньатъття щось мальувати цецьікаве  
 14 мальувати щозь гарне цецьікаве заньатъття  
 15 гарне щось мальувати цецьікаве заньатъття  
 16 щозь гарне мальувати цецьікаве заньатъття  
 17 щось мальувати гарне заньатъття цецьікаве  
 18 щозь гарне мальувати заньатъття цецьікаве  
 19 мальувати цецьікаве заньатъття гарне щось  
 20 цецьікаве мальувати заньатъття гарне щось  
 21 мальувати цецьікаве заньатъття щозь гарне  
 22 цецьікаве мальувати заньатъття щозь гарне

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

22

? це  
 ? цьі  
 ? ка 6  
 ? ве  
 ? шчо 6  
 ? зьгар 6  
 ? не  
 ? ма 3  
 ? льу  
 ? ва 6  
 ? ти  
 ? за  
 ? нья 6  
 ? тьтѣа

1 цецьікаве шчозьгарне мальувати заньатѣтѣа  
 2 шчозьгарне цецьікаве мальувати заньатѣтѣа  
 3 заньатѣтѣа шчозьгарне мальувати цецьікаве  
 4 шчозьгарне заньатѣтѣа мальувати цецьікаве  
 5 заньатѣтѣа шчозьгарне цецьікаве мальувати  
 6 шчозьгарне заньатѣтѣа цецьікаве мальувати  
 7 шчозьгарне цецьікаве заньатѣтѣа мальувати  
 8 мальувати шчозьгарне цецьікаве заньатѣтѣа  
 9 шчозьгарне мальувати цецьікаве заньатѣтѣа  
 10 шчозьгарне мальувати заньатѣтѣа цецьікаве  
 11 цецьікаве мальувати заньатѣтѣа шчозьгарне  
 12 мальувати цецьікаве заньатѣтѣа шчозьгарне

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

8

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

мальувати шчозьгарне цецьікаве заньатьтя

Please, note that prolongation of the consonants is encoded and displayed as their repeating, including the case of palatalized or semi-palatalized consonants the groups with the soft sign

repeat ("заньатьтя"). Besides, pay attention to the fact that during the permutations of the words there were some phonetic changes – "щось гарне" became "шчозь гарне". If we had took the first word and had connected with the other with no phonetic change should be, then the program would pass this and this would introduce phonetic deviation as a result. Nevertheless, taking into account the proportions of the durations the deviation between the voiced and voiceless corresponding (or 'approximately corresponding' as in case of r-x) consonants is about no more than 11%. Several potential such deviations per one line do not change the rhythmic picture too much, and for the properties with 'a' in signage do not influence at all.

## Ideas that can help

If you would like to carry some exploration of the line, then different schemes can be used for the analysis. The programs give a lot of opportunities so the appropriate are needed. Here, there are several advices:

1. You can begin the analysis by sequentially specifying properties of different character, e. g. for 2 syllables (using the first command line argument as N\_0.0), 3 syllables (using the first argument as 0\_N), w / x, then 5- or 6-syllables options for the properties, you can specify also your custom ones.
2. Usage of the properties 'a' type needs more time but it allows to specify more characteristics and, moreover, gives advances in syllabotonic versing.
3. An interesting information you can obtain using also the details (see further).

4. And much more even more creative options, at last with the programs usage the inner sense of the language must develop more and more and the human can learn to create beautiful texts.

## Details in the multiple properties mode

If you use a multiple properties mode (+m ...-m), then for the obtaining the additional information there can be more details displayed. To get more detailed output, you should specify as the command line arguments outside the groups of +m ...-m and +a ...-a the arguments +v number 1, 2 or 3. While so, the program before the output to the display the common options for all specified properties and parameters will output additionally also sorted out by String (in case of 1), by the intermediate value (in case of 2), or by the last value calculated (in case of 3) of the type Double (sorting by the last one is the basis of the work of the algorithm in most cases).

Example:

```
lineVariantsG3 +m as01 3 w02 10 -m +s 1 +v 1 +и малювати щось гарне це цікаве заняття
1
? ма 3
? льу
? ва 6
? ти
? шчозь 6
? гар 6
? не
? це
? цьі
? ка 6
? ве
? за
? нья 6
? тьтѡа
гарне заняття щось малювати цецїкаве ->578.0->578.0
```



гарне заняття щось цецікаве малювати ->578.0->578.0  
 гарне щось малювати цецікаве заняття ->560.0->560.0  
 гарне щось цецікаве малювати заняття ->578.0->578.0  
 заняття гарне щось малювати цецікаве ->578.0->578.0  
 заняття гарне щось цецікаве малювати ->578.0->578.0  
 заняття щось гарне малювати цецікаве ->578.0->578.0  
 заняття щось гарне цецікаве малювати ->578.0->578.0  
 малювати цецікаве заняття гарне щось ->544.0->544.0  
 малювати цецікаве заняття щось гарне ->544.0->544.0  
 малювати щось гарне цецікаве заняття ->560.0->560.0  
 цецікаве малювати заняття гарне щось ->544.0->544.0  
 цецікаве малювати заняття щось гарне ->544.0->544.0  
 цецікаве щось гарне малювати заняття ->578.0->578.0  
 щось гарне заняття малювати цецікаве ->578.0->578.0  
 щось гарне заняття цецікаве малювати ->578.0->578.0  
 щось гарне малювати заняття цецікаве ->560.0->560.0  
 щось гарне малювати цецікаве заняття ->560.0->560.0  
 щось гарне цецікаве заняття малювати ->578.0->578.0  
 щось гарне цецікаве малювати заняття ->578.0->578.0  
 щось малювати гарне заняття цецікаве ->560.0->560.0  
 щось цецікаве гарне заняття малювати ->578.0->578.0

гарне малювати щось заняття цецікаве ->0.6328125->0.6328125  
 гарне малювати щось цецікаве заняття ->3.65625->3.65625  
 заняття малювати щось цецікаве гарне ->0.4921875->0.4921875  
 заняття щось цецікаве гарне малювати ->0.421875->0.421875  
 малювати гарне заняття цецікаве щось ->3.0625->3.0625  
 малювати гарне цецікаве щось заняття ->0.6640625->0.6640625

малювати щось гарне заняття цецікаве ->0.53125->0.53125  
 малювати щось заняття гарне цецікаве ->3.0625->3.0625  
 малювати щось заняття цецікаве гарне ->0.4375->0.4375  
 малювати щось цецікаве гарне заняття ->0.53125->0.53125  
 цецікаве гарне заняття малювати щось ->0.4375->0.4375  
 цецікаве гарне малювати щось заняття ->0.6328125->0.6328125  
 цецікаве щось гарне малювати заняття ->0.4375->0.4375  
 цецікаве щось заняття малювати гарне ->0.4375->0.4375  
 щось гарне малювати заняття цецікаве ->0.4375->0.4375  
 щось заняття малювати гарне цецікаве ->3.71875->3.71875  
 щось цецікаве гарне заняття малювати ->0.40625->0.40625

шчозь гарне малювати заньатьтя цецьікаве  
 шчось цецьікаве гарне заньатьтя малювати  
 цецьікаве шчозь гарне малювати заньатьтя

While doing so, the arrows indicate in every case the flow of calculations (logics) of the values of properties and the grouping order corresponds to the order in which the properties are specified in the command line. Here, the sorting is done by the String itself (+v 1), the String's are sorted by ascending, and contrary – the numbers (in cases of +v 2 / +v 3) would be sorted by descending.

## Work with program unconcatUkr

While execution of the lineVariantsG3, rewritePoemG3 there become present some word concatenations for the preserving the minima grammar and better text understanding. Some of them are typical and they can be easily distinguish from the usual (not concatenated) words. Therefore, for the speeding up the text post-editing with the programs usage you can use also the other program unconcatUkr.

```
unconcatUkr 1 <path to the file with the concatenated words> [<path to the new file>]  
unconcatUkr 2 <path to the file with the concatenated words> [<path to the new file>]  
unconcatUkr -i
```

are the three principal ways to utilize it.

The first one is the 'safest', it uses just those unconcatenations, which almost for sure won't lead to the mistakes in the words. The second one is more 'risky' (aggressive) but has more effect, nevertheless, use it with caution and be aware of its possible consequences (some not so widely spread words will be wrongly splitted while the execution).

By default, if the number is not specified, then it is equal to 1.

# Working with propertiesTextG3 (and distributionTextG)

## Option I (lines only)

Verify that the folder (directory) where cabal installed the executables of the programs is available for search in the PATH environment variable.

Then enter the command at the command prompt (or terminal):

```
propertiesTextG3 <the first argument> [<WX argument>] [<whether to use  
'growing lines'>] <file with Ukrainian text> <control of the number  
of intervals> <control whether to print also text string> <control the  
division of text into lines> <property type>
```

and press Enter.

You will see something like:

```
propertiesTextG3 2.1_3.0 sadok.txt s 1 0 03z  
4  
8 8 8 1.0029 1.0029 1.0000 1.00143062 2 4  
Тарас ШЕВЧЕНКО  
Вказематі  
108 113 244 1.0425 2.2559 2.1640 0.64036546 3 1
```

Садок вишневий колохати  
13 29 29 2.3234 2.3234 1.0000 1.39819820 3 4  
Хрущі надвишнями гудуть  
5 12 32 2.2321 6.0223 2.6980 0.63572791 3 1  
Плугатарі зплугами йдуть  
108 108 110 1.0000 1.0182 1.0182 0.99097861 3 1  
Співають ідуци дівчата  
12 13 49 1.0984 4.0438 3.6814 0.43556382 3 1  
Аматері вечерять ждуть  
27 29 110 1.0685 4.0428 3.7835 0.42378641 3 1  
Сем'я вечера колохати  
12 12 27 1.0000 2.2500 2.2500 0.61538462 3 1  
Вечірня зіронька встає  
3 14 14 4.3608 4.3608 1.0000 1.62692308 3 4  
Дочка вечерять подає  
248 508 508 2.0437 2.0437 1.0000 1.34289977 3 4  
Амати хоче научати  
1 13 13 24.6349 24.6349 1.0000 1.92198142 3 4  
Так соловейко недає  
108 112 113 1.0346 1.0438 1.0089 1.01242119 3 4  
Поклала мати колохати  
1 13 13 25.1071 25.1071 1.0000 1.92339261 3 4  
Маленьких діточок своїх  
3 4 4 1.1995 1.3353 1.1132 1.02728205 3 3  
Сама заснула колойіх  
108 108 248 1.0000 2.2937 2.2937 0.60721063 4 1  
Затихло все тільки дівчата  
3 13 13 4.0000 4.0000 1.0000 1.60000000 2 4  
Тасоловейко незатих

Міжтравня  
С-Петербург

Numeric columns have the same value for both options. The difference is that in the second case the statistics on the whole text is more important from the point of view of the researcher than for each line in particular.

Column I is the minimum possible value of the selected property for the given data among all possible variants of permutations of words in line;

Column II is the actual value of the selected property for the specified data in the row, the one that is implemented in this particular version of the row;

Column III is the maximum possible value of the selected property for the given data among all possible variants of word permutations in line;

Column IV is the ratio of the value of the property for a given row and its minimum value for the words that make up the row; a numeric, which is not less than 1.0;

Column V is the ratio of the maximum value of the property for the words of this line and its minimum value, which consists of line; a number that is not less than 1.0 and not less than the number in column IV;

Column VI is the ratio of the maximum value of a property for a given row and its actual value; a number that is not less than 1.0 and not greater than the value in the V column;

Column VII is the ratio of the actual value of the property to the arithmetic mean (half-sum) of the maximum and minimum values for all possible permutations of words for a given data; a number that is displayed with the full calculated number of characters after dots; is important for further statistics for the whole text;

Column VIII is the number of words in a line, some of which may consist of several connected Ukrainian words;

Column IX is the number of the interval (starting with 1), which includes the actual value of the property for the specified data;

Further to the right - if <control whether to print also a line of text> as '1', then a line of text that is being analyzed is displayed here; otherwise, it is not displayed.

## **Option II - statistics throughout the text (+ possibly lines)**

Verify that the folder (directory) where cabal installed the executables of the programs is available for search in the PATH environment variable.

Then enter the command in the terminal:

```
propertiesTextG3 <the first argument> [<WX argument>] [<whether to use
'growing lines'>] <file with Ukrainian text> <control of the number of
intervals> <control whether to print also text string> <control the
division of text into lines > <property type> | distributionTextG
<same row argument> <whether also display ordinal data>
```

and press Enter. In Unix-like operating systems, the vertical line (highlighted in red) is used to create pipelines in the shell terminal; for Windows:

```
PowerShell -Command "propertiesTextG3 <the first argument> [<WX argument>]
[<whether to use 'growing lines'>] <file with Ukrainian text> <quantity
control intervals> <control whether to print also text string> <control
the division of text into lines> <property type> | distributionTextG
< same row argument > <or also display ordinal data >'
```

You will see something like:

```
propertiesTextG3 2.1_3.0 sadok.txt s 1 0 02y | distributionTextG s 1 +W
```

|                          |     |     |   |        |        |        |            |   |   |
|--------------------------|-----|-----|---|--------|--------|--------|------------|---|---|
| 3                        | 4   | 3   | 5 | 1.0000 | 1.7000 | 1.7000 | 0.74074074 | 2 | 1 |
| Тарас ШЕВЧЕНКО           |     |     |   |        |        |        |            |   |   |
| Вказематі                |     |     |   |        |        |        |            |   |   |
| 108                      | 108 | 142 |   | 1.0000 | 1.3111 | 1.3111 | 0.86538462 |   | 3 |
| Садок вишневий колохати  |     |     |   |        |        |        |            |   |   |
| 4                        | 34  | 34  |   | 8.5180 | 8.5180 | 1.0000 | 1.78987107 | 3 | 4 |
| Хрущі надвишнями гудуть  |     |     |   |        |        |        |            |   |   |
| 3                        | 25  | 25  |   | 8.1680 | 8.1680 | 1.0000 | 1.78184991 | 3 | 4 |
| Плугатарі зпругами йдуть |     |     |   |        |        |        |            |   |   |

104

|                            |     |     |         |         |        |            |   |   |
|----------------------------|-----|-----|---------|---------|--------|------------|---|---|
| 108                        | 108 | 142 | 1.0000  | 1.3111  | 1.3111 | 0.86538462 | 3 | 1 |
| Співають ідучи дівчата     |     |     |         |         |        |            |   |   |
| 3                          | 4   | 25  | 1.3111  | 8.1680  | 6.2298 | 0.28601900 | 3 | 1 |
| Аматері вечерять ждуть     |     |     |         |         |        |            |   |   |
| 108                        | 245 | 245 | 2.2694  | 2.2694  | 1.0000 | 1.38827528 | 3 | 4 |
| Сем'я вечеря колохати      |     |     |         |         |        |            |   |   |
| 3                          | 34  | 34  | 11.1680 | 11.1680 | 1.0000 | 1.83563445 | 3 | 4 |
| Вечірня зіронька встає     |     |     |         |         |        |            |   |   |
| 3                          | 34  | 34  | 11.1680 | 11.1680 | 1.0000 | 1.83563445 | 3 | 4 |
| Дочка вечерять подає       |     |     |         |         |        |            |   |   |
| 108                        | 277 | 277 | 2.5611  | 2.5611  | 1.0000 | 1.43837754 | 3 | 4 |
| Амати хоче научати         |     |     |         |         |        |            |   |   |
| 3                          | 3   | 13  | 1.0000  | 4.3111  | 4.3111 | 0.37656904 | 3 | 1 |
| Так соловейко недає        |     |     |         |         |        |            |   |   |
| 108                        | 142 | 245 | 1.3111  | 2.2694  | 1.7309 | 0.80203908 | 3 | 1 |
| Поклала мати колохати      |     |     |         |         |        |            |   |   |
| 3                          | 3   | 25  | 1.0000  | 8.1680  | 8.1680 | 0.21815009 | 3 | 1 |
| Маленьких діточок своїх    |     |     |         |         |        |            |   |   |
| 3                          | 12  | 25  | 4.0000  | 8.1680  | 2.0420 | 0.87260035 | 3 | 2 |
| Сама заснула колойіх       |     |     |         |         |        |            |   |   |
| 27                         | 29  | 242 | 1.0778  | 8.9778  | 8.3299 | 0.21603563 | 4 | 1 |
| Затихло все тільки дівчата |     |     |         |         |        |            |   |   |
| 4                          | 4   | 12  | 1.0000  | 3.0508  | 3.0508 | 0.49372385 | 2 | 1 |
| Тасоловейко незатих        |     |     |         |         |        |            |   |   |
| Міжітравня                 |     |     |         |         |        |            |   |   |
| С-Петербург                |     |     |         |         |        |            |   |   |

---

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 9 | 1 | 0 | 6 |



|       | 56.25%         | 6.25% | 0.00% | 37.50% |
|-------|----------------|-------|-------|--------|
|       | 0.9879+-0.6062 | 0     | 16    |        |
| 2     | 3              | 4     | 5     | 6      |
| 2     | 13             | 1     | 0     | 0      |
| ***** |                |       |       |        |
| 2     | .              | .     | .     | .      |
| 6     | 1              | .     | 6     | .      |
| 1     | .              | .     | .     | .      |
| .     | .              | .     | .     | .      |
| .     | .              | .     | .     | .      |
| ~~~~~ |                |       |       |        |
| 2     | 0              | 0     | 0     | 0      |
| 6     | 1              | 0     | 6     | 0      |
| 1     | 0              | 0     | 0     | 0      |
| 0     | 0              | 0     | 0     | 0      |
| 0     | 0              | 0     | 0     | 0      |
| 0     | 0              | 0     | 0     | 0      |

There will be no color highlighting, just different types of statistics are grouped here (see the table below for semantics).

1. Red colour – Decimal fraction with error – The arithmetic mean (mathematical expectation) of all numbers in column VII statistics for rows, plus or minus standard quadratic deviation; in the case of the selected properties "y0" - often a number close to 1.0; may contain a rounding error caused by finding the sum of numbers with floating point. In case all lines of text excluded from the analysis (see: explanation for orange) only one line of text - signal inscription is displayed instead of matrices and further information:

"1,000+-0,000!",

where " " means a tab character. This label means that the specified text is not suitable for program analysis as well as that any data that is gotten during the execution of the program (could be derived from this text and related texts) should be analyzed

as follows, so as not to refute the conclusions made on based on the whole set of texts. Simply speaking, in this case, you can not take into account the text, because with the right approach to the analysis and interpretation of data it should not break the results.

2. Blue colour – Natural numbers – Row distribution matrix by number of words and intervals; matrix element in the k-th row and j-th column - the number of rows for which the value of the selected property with the specified data falls in the interval with the number j (numbering starts with 1), which is equal to matrix column number, and contains in the line k words (words or combinations thereof that are displayed merged to comply with minimum grammar rules in data analysis and output), the number k is in the range between 2 and 7 inclusive (rows that are analyzed for the matrix contain from 2 to 7 words (or written together their concatenations)). Thus, the matrix always has 6 rows, and the number of columns will depend on entered and available data. In fact, this matrix replaces the graphical two-dimensional data distribution. It is displayed twice, one after the other delimited by tildes. In the first case zero values are not displayed, instead there are dots. This is an element of data visualization, which allows you to better 'see' how a distribution looks like where numeric values correspond to the 'height' on the distribution graph (the value of the discrete two-dimensional function distribution). In the second case in place of dots there are the corresponding values, which are all equal to 0. Data for the matrix are obtained from VIII and IX columns of statistics by rows.
3. Orange colour – Non-negative integers – Total number of rows. The first number on the left is the number of rows that are excluded from the analysis for the matrix, because they have little data (1 or less words). Equality 0 means that all lines displayed on the screen participate in the analysis for formation of a distribution matrix. Number on the right - the total number of lines in the text that are displayed and analyzed (including those rows that are listed on the left).
4. Green colour – Percents – Distribution of the total number of rows by intervals. The sum of percentage values naturally equal to 100%. Interval numbers inscribed above the corresponding percentage values. For example, the inscription in these three lines type:
 

1 2  
10 15  
40% 60%

 means that of the total number of rows, which can be analyzed using the program (contain enough data), 40% is accounted for by the first interval (with a smaller value properties), and 60% - by the second one (according to greater value of the property).

That is such lines 10 and 15, respectively. All intervals are equal in size, but can have different numbers of rows. This is a simple one-dimensional distribution, it is possible to build a histogram.

5. Yellow colour – Natural numbers – Interval numbers. The countdown begins with 1. Further below they correspond to the number of lines, values of the properties for which according to the data falls into the corresponding number interval.
6. Brown colour – Natural numbers – Number of words in lines. Lies within 2 up to 7 inclusive (if there are less words, then the line gets a value of 1.0 and is removed from analysis program for the matrix). Under them - the corresponding values of the number of such lines. 0 corresponds to the case of the absence of lines with the number of words (or concatenations that are displayed as one word).

Careful study of these data allows us to draw certain conclusions about the text, their totality, the model and language itself.

## Statistics mode by multiple properties (+m ... -m)

Now, as for the program lineVariantsG3, you can use the mode of multiple properties. To do this, instead of one property it is possible to specify multiple ones in the block selected by delimiters +m ... -m.

In this case, the program will display something like:

У такому разі програма виведе на екран щось на зразок:

```
propertiesTextG3 sadok.txt s 1 0 +m y0 0y 02y 03y y2 y3 yy3 -m
2          4          2          1          2          4          4          1          Тарас ШЕВЧЕНКО
          Вказематі
3          4          4          1          2          4          2          1          Садок вишневий колохати
3          4          1          4          4          4          4          4          Хрущі надвишнями гудуть
3          4          4          4          1          4          1          1          Плугатарі зплугами йдуть
```

|   |   |   |   |   |   |   |   |   |                            |
|---|---|---|---|---|---|---|---|---|----------------------------|
| 3 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | Співають ідучи дівчата     |
| 3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | Аматері вечерять ждуть     |
| 3 | 1 | 3 | 4 | 1 | 3 | 1 | 2 | 2 | Сем'я вечеря колохати      |
| 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | Вечірня зіронька встає     |
| 3 | 4 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | Дочка вечерять подає       |
| 3 | 4 | 4 | 4 | 1 | 4 | 2 | 1 | 1 | Амати хоче научати         |
| 3 | 4 | 1 | 1 | 4 | 1 | 4 | 3 | 3 | Так соловейко недає        |
| 3 | 4 | 4 | 2 | 1 | 2 | 4 | 1 | 1 | Поклала мати колохати      |
| 3 | 1 | 4 | 1 | 4 | 1 | 2 | 4 | 4 | Маленьких діточок своїх    |
| 3 | 1 | 4 | 2 | 4 | 1 | 2 | 4 | 4 | Сама заснула колойіх       |
| 4 | 3 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | Затихло все тільки дівчата |
| 2 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | Тасоловейко незатих        |

Міжітравня  
С-Петербург

In this case, you do not need to use the `distributionTextG` program, because its behaviour is not defined here.

The first column (highlighted in color here) - the number of words in the appropriate rows; the columns follow in the order in which they appear marked in the block of several properties, respectively - the numbers of the intervals, which include the values of the corresponding properties. The first number on the right (highlighted in red and the only one in its line) is the number of intervals for each property (they are all the same). Eight columns in this case to text records - means that there were 7 (= 8 - 1) given properties in the block.

### Several properties mode +m for `distributionTextG`

In such a case, it is possible to use the program `distributionTextG` by adding to command line the parameter +m:

```
propertiesTextG3 sadok.txt s 1 0 +m y0 0y 02y 03y y2 y3 yy3 -m | distributionTextG s 1 +W +m
```

4

```
2 4 2 1 2 4 4 1 Тарас ШЕВЧЕНКО  
Вказематі
```

3 4 4 1 2 4 2 1 Садок вишневий колохати  
 3 4 1 4 4 4 4 4 Хрущі надвишнями гудуть  
 3 4 4 4 1 4 1 1 Плугатарі зплугами йдуть  
 3 2 4 1 1 1 1 1 Співають ідучи дівчата  
 3 2 2 1 1 1 1 1 Аматері вечерять ждуть  
 3 1 3 4 1 3 1 2 Сем'я вечеря колохати  
 3 3 4 4 4 4 4 4 Вечірня зіронька встає  
 3 4 2 4 4 4 4 4 Дочка вечерять подає  
 3 4 4 4 1 4 2 1 Аматери хоче научати  
 3 4 1 1 4 1 4 3 Так соловейко недає  
 3 4 4 2 1 2 4 1 Поклала мати колохати  
 3 1 4 1 4 1 2 4 Маленьких діточок своїх  
 3 1 4 2 4 1 2 4 Сама заснула колоїх  
 4 3 1 1 2 1 2 2 Затихло все тільки дівчата  
 2 4 4 1 1 1 1 1 Тасоловейко незатих

Міжтравня  
 С-Петербург

-----  
 1 2 3 4  
 3 2 2 9  
 18.75% 12.50% 12.50% 56.25%  
 0 16  
 2 3 4 5 6 7  
 2 13 1 0 0 0

\*\*\*\*\*  
 . . . 2  
 3 2 1 7  
 . . 1 .  
 . . . .

. . . .  
. . . .

0 0 0 2  
3 2 1 7  
0 0 1 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

---

---

1 2 3 4  
3 3 1 9  
18.75% 18.75% 6.25% 56.25%  
0 16  
2 3 4 5 6 7  
2 13 1 0 0 0

\*\*\*\*\*  
. 1 . 1  
2 2 1 8  
1 . . .  
. . . .  
. . . .  
. . . .

---

---

0 1 0 1  
2 2 1 8  
1 0 0 0  
0 0 0 0

0 0 0 0  
0 0 0 0

---

---

1 2 3 4  
8 2 0 6  
50.00% 12.50% 0.00% 37.50%

0 16  
2 3 4 5 6 7  
2 13 1 0 0 0

\*\*\*\*\*

2 . . .  
5 2 . 6  
1 . . .  
. . . .  
. . . .  
. . . .

2 0 0 0  
5 2 0 6  
1 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

---

---

1 2 3 4  
7 3 0 6  
43.75% 18.75% 0.00% 37.50%

0 16  
2 3 4 5 6 7  
2 13 1 0 0 0

\*\*\*\*\*

1 1 . .  
6 1 . 6  
. 1 . .  
. . . .  
. . . .  
. . . .

~~~~~

1 1 0 0
6 1 0 6
0 1 0 0
0 0 0 0
0 0 0 0
0 0 0 0

=====

1 2 3 4
7 1 1 7
43.75% 6.25% 6.25% 43.75%
0 16
2 3 4 5 6 7
2 13 1 0 0 0

1 . . 1
5 1 1 6
1 . . .

. . . .
. . . .
. . . .

~~~~~  
1 0 0 1  
5 1 1 6  
1 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

-----  
1 2 3 4  
5 5 0 6  
31.25% 31.25% 0.00% 37.50%  
0 16  
2 3 4 5 6 7  
2 13 1 0 0 0

\*\*\*\*\*  
1 . . 1  
4 4 . 5  
. 1 . .  
. . . .  
. . . .  
. . . .

~~~~~  
1 0 0 1
4 4 0 5
0 1 0 0

0 0 0 0
0 0 0 0
0 0 0 0

1 2 3 4
8 2 1 5
50.00% 12.50% 6.25% 31.25%

0 16
2 3 4 5 6 7
2 13 1 0 0 0

2 . . .
6 1 1 5
. 1 . .
. . . .
. . . .
. . . .

~~~~~  
2 0 0 0  
6 1 1 5  
0 1 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

---

---

In such a case, the program will output the possible statistics by every property starting count from left to right. Here, there are 7 properties, so there is output for 7 distributions.

## White String Mode

The program can also now use 'white lines' mode, which means that lines that contain fewer words than needed to ensure the existence of at least two variants of the string, do not display statistics and it is not included in the overall result. Then in the case of one metric and the use of the distributionTextG program, you need to call the latter with an additional one argument +W (means whitelines).

For example, in this case you will see:

```
propertiesTextG3 sadok.txt s 1 0 03y +b | distributionTextG s 1 +W
```

```

4
Тарас ШЕВЧЕНКО
Вказематі
52      52      81      1.0000      1.5577      1.5577      0.78195489      3      1
Садок вишневий колохати
4      14      14      3.2040      3.2040      1.0000      1.52426261      3      4
Хрущі надвишнями гудуть
1      1      4      1.0000      2.7692      2.7692      0.53061224      3      1
Плугатарі зплугами йдуть
36      36      40      1.0000      1.1111      1.1111      0.94736842      3      1
Співають ідучи дівчата
1      1      1      1.4444      1.4444      1.0000      1.18181818      3      4
Аматері вечерять ждуть
36      36      52      1.0000      1.4444      1.4444      0.81818182      3      1
Сем'я вечеря колохати
14      14      14      1.0000      1.0000      1.0000      1.00000000      3      2
Вечірня зіронька встає
1      14      14      14.2400      14.2400      1.0000      1.86876640      3      4
Дочка вечерять подає
37      37      37      1.0000      1.0000      1.0000      1.00000000      3      2
Амати хоче научати
1      4      4      4.0000      4.0000      1.0000      1.60000000      3      4

```

|                            |    |     |         |         |        |            |   |   |  |
|----------------------------|----|-----|---------|---------|--------|------------|---|---|--|
| Так соловейко недає        |    |     |         |         |        |            |   |   |  |
| 36                         | 40 | 40  | 1.1111  | 1.1111  | 1.0000 | 1.05263158 | 3 | 4 |  |
| Поклала мати колохати      |    |     |         |         |        |            |   |   |  |
| 1                          | 11 | 11  | 11.2400 | 11.2400 | 1.0000 | 1.83660131 | 3 | 4 |  |
| Маленьких діточок своїх    |    |     |         |         |        |            |   |   |  |
| 1                          | 4  | 4   | 4.4444  | 4.4444  | 1.0000 | 1.63265306 | 3 | 4 |  |
| Сама заснула колоїїх       |    |     |         |         |        |            |   |   |  |
| 9                          | 52 | 100 | 5.7778  | 11.1111 | 1.9231 | 0.95412844 | 4 | 2 |  |
| Затихло все тільки дівчата |    |     |         |         |        |            |   |   |  |
| Тасоловейко незатих        |    |     |         |         |        |            |   |   |  |
| Міжітравня                 |    |     |         |         |        |            |   |   |  |
| С-Петербург                |    |     |         |         |        |            |   |   |  |

---

|                |    |        |       |        |   |
|----------------|----|--------|-------|--------|---|
| 1              | 2  | 3      | 4     |        |   |
| 4              | 3  | 0      | 7     |        |   |
| 28.57%         |    | 21.43% | 0.00% | 50.00% |   |
| 1.1949+-0.4206 |    | 0      | 14    |        |   |
| 2              | 3  | 4      | 5     | 6      | 7 |
| 0              | 13 | 1      | 0     | 0      | 0 |

\*\*\*\*\*

|   |   |   |   |
|---|---|---|---|
| . | . | . | . |
| 4 | 2 | . | 7 |
| . | 1 | . | . |
| . | . | . | . |
| . | . | . | . |

---

|   |   |   |   |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 4 | 2 | 0 | 7 |

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

---

The 'white' lines here are shown as some space where there would not be it otherwise.

## Reduced Set of Permutations Mode

In the reduce set of permutations mode the both programs `propertiesTextG3` and `distributionTextG` can work. In such a case as an command line argument must be specified «+p 1 or 2» for the first one and just «+p» for the second one. In such a case, the statistics is displayed so as there are the possible number of the words in the line between 2 and 10 inclusively.

## Fixed-line statistics (+b)

If you specify as one of the command line arguments for `propertiesTextG3` characters +b, the program will calculate all statistics, as if the last word is fixed by constraint and does not move. In fact, in this case, the meaning of this symbol (argument) is similar to the `lineVariantsG3` program. It is necessary to remember that it narrows a range of admissible values of properties and at invariable quantity of lines changes distributions inside the intervals.

## Control the number of intervals

There are three possible cases:

- “s” – the number of intervals will be determined by the well-known Sturge’s rule where the number of tests will be equal to the resulting number of lines;

- “l” – the number of intervals will be determined on the recommendation of V. P. Levinsky (see: Опря А. Т. Статистика (модульний варіант з програмованою формою контролю знань). – Навч. посіб. – К.: Центр учбової літератури, 2012. – 448 с. ISBN 978-611-01-0266-7. С. 60);
- the number of intervals will be a natural number (must be greater than 1, although this is not checked);
- something else - will be used 9.

## **You can also control whether to print lines of the text**

If this argument is 1, then to the right of the numerical data of the ordinal statistics a line that is analyzed will also be displayed (already converted for analysis). Otherwise the string will not be displayed in the output.

## **Control the division of text into lines**

If you specify 1 here, the text will firstly be grouped into one line, and then divided into lines by the method of division in half (by the number of words or their concatenations) until the length of all lines is less than 8 words or their combinations. Foreign characters will be filtered.

If set to 0, the text will be parsed after filtering out extraneous characters (approximately) in the lines that were originally.

## **Whether to use ‘growing lines’ mode**

If among the command line arguments you specify “+g ab”, where a, b are some digits except 0, then the ‘growing lines’ mode is used. This means that the text is transformed in a way so that at first the maximum number of words in lines is no greater than the second digit and then the lines are grouped so that the number of words in every line is close to the first digit if it can be achieved by concatenation of the lines into the single one consequently. In other words, there will be done some transformation of the text by lines to make the number of words in every line closer to the first digit and no more than 7 (for the last one the quotient is used).

In such a case, the control of division of text into lines plays less role, but it can influence if it is equal to 1. For example, “+g 73” as a command line argument means that after the application of the division of the lines all them are partitioned so that in every line there are at first no more than 3 words or their concatenations and then the lines are concatenated so that in every of them there are a number of the words close to 7 (and no more).

Please, note that starting from the version 0.16.0.0 there is a space between the +g and the two-digits number. Earlier there has not been any.

## **The same argument for the number of rows**

It means there should be the same value, as in the place of control of the number of intervals.

## **Whether to display serial data as well**

Here it is necessary to put 1 so that the program prints all statistics (at first by lines, and then the general on the text), otherwise only the general text is printed.

## **Selective analysis of text by lines**

If you execute a command

```
propertiesTextG3 <path to the file with Ukrainian text for analysis> @n
```

then the text from the file will be displayed on the screen and will be shown with the numbers of all lines to the left of the lines themselves, separated from the text using the tab character (displayed as a space with a non-constant width, which depends on the system settings). Then you can run the same program (you can do without it, but you can specify other numbers than the program will consider) for analysis the selected rows. To do this, to the commands `propertiesTextG3`, in addition to the last mentioned, anywhere in the command line to the vertical bar (up to pipeline) add the first and last line numbers, separated by a colon (without any other

characters, including spaces). You can specify several such pairs, the information will be displayed in the same order. If some line numbers will occur several times, they will be displayed (if this option is specified) and analyzed several times (probably, due to the laziness of Haskell programming language, they can use (if they are not garbage collected till the time of re-usage) the once calculated and memoized results). If text break control is specified on lines equal to 1, then the program will combine and analyze the lines whose numbers were specified and correspond to the numbers when outputting the command from @n.

All this allows you to focus on the text or only part of it.



# Working with rewritePoemG3

Verify that the folder (directory) where cabal installed the executables of the programs is available for search in the PATH environment variable.

The program rewritePoemG3 starting from the version 0.12.0.0 of the package can work in several modes: in the multiple properties mode, in the single property mode, in the comparative mode.

## Multiple properties mode (+m ...-m)

If among the command line arguments there is specified a group of arguments inside the frame of +m and -m delimiters and inside it there are several properties in their encoding (see: Types of properties) delimiting each of them by the space from the previous and the next ones and if there are not specified "+c" as one of the command line arguments, then the program will work in the multiple properties mode (if the property inside the frame is just one solely, then the program will run in the single property mode, see below).

The syntax of the program work in such a case is the following:

```
rewritePoemG3 <first argument> <file with Ukrainian text> [<whether to use  
"growing lines">] +m <types of properties> -m <numeric arguments>
```

Upon successful completion of the program (there should be no messages) in the same folder (directory) as the file with the text that is overwritten, there must be files with the additional ending '.new.txt' and the prefixes delimited with the dot with the properties encoding. It is there that the converted texts (for example, a poem) are written to according to the input data.

The entered data applies to the entire text, to each line of text in particular (after its preliminary processing by the program).

## Single property mode

If you do not specify among the command line arguments a frame with the +m and -m delimiters, and if there is no "+c" specification either, then the program will work in the single property mode (it is also applicable in the case of just single property specified in the multiple properties mode).

The syntax of the program work in such a case is the following:

```
rewritePoemG3 <first argument > [<whether to grow lines>] <Ukrainian text file>
<property type> <numerical arguments>
```

You will see something like the following:

```
rewritePoemG3 10.0_1.2 sadok.txt yy 5 1 2
```

Upon successful completion of the program (there should be no messages) in the same folder (directory) as the file with the text that is overwritten, there must be a file with the additional ending '.new.txt' and the prefix delimited with the dot with the property encoding. It is there that the converted text (for example, a poem) is written according to the input data.

The entered data applies to the entire text, to each line of text in particular (after its preliminary processing by the program).

## More complex usage

While using the multiple properties mode or the single property mode the numeric arguments have much the same meaning as for the lineVariantsG3 program.

Numeric arguments, if specified, have the following meaning.

The first numeric argument is the number of intervals into which the interval between the minimum and maximum value of the property for this line. If not specified, it is considered equal to 1. A value of 0 does not allow other numeric arguments to further change the result of the work of the program.

All subsequent numeric arguments (if specified, otherwise no permutations occur) are interval numbers that will be swapped with the interval with the maximum values of the property. This allows you to change the structure of the data that is displayed as a result of

the program and see the internal (not maximum) items. For example, the numeric arguments 6 1 4 (in this order) will mean that during program execution the interval between the maximum and minimum value of the property will be divided into 6 equal intervals, with the elements that are in the first and fourth, counting from the minimum (interval number 1) will be moved to the maximum number (and property values) interval, and then the line with the maximum value of the property is written to the output file. Values that were in the maximum interval will be moved to the interval with the lowest number among those that are moved to maximum.

## **Comparative mode of operation (+c)**

rewritePoemG3 can also be run in so-called 'comparative' mode, when it offers strings (one after the other) of several (no more than 7) specified files and writes the selected (or blank line, if none is selected) to the last file (except those ones). So from several files by their comparison by lines you can create a new one. It also allows you to run the program with different properties in the multiple properties mode, then run it in the comparative mode on the received files and create in a fairly easy way their combinations - new variants.

Note: If you plan to get more 'hints' and recommendations from the program, it is probably easier (and better) to apply an interactive mode of the lineVariantsG3 program with several properties instead, or even the recursive mode of the multiple sources mode of the lineVariantsG3.

To operate in the comparative mode, use the following command:

```
rewritePoemG3 +c <files to read text variants from> <final file>
```

# Types of properties

One of the principles of the program is to search among the text options for those for which the maximum is the value of a function called 'property' of the text (just simply: property) and is a specific property for lines. The user can choose the property that will be used during program operation (this is done in the command line once during program operation by the given (or absent respectively) command line argument). The command line argument of the program call can be:

- 'y0' - the first property in time, based on 'periods of uniqueness'. The idea is to estimate the number of sounds, or pauses, or phonetic phenomenae (palatalization of consonants), which are between successive appearances of each sound not in one, but in different words, and the total sum of such distances for different words is sought. The greater value for the text corresponds to that one, in which each new word offers more and more new sounds, (presumably) depending on the average number of sounds in the "period of uniqueness" it may be easier to pronounce; a lower value - on the contrary - to the text in which there are more returns in a new word to the sounds of the previous ones, thus, there may be more repeaters, emphasizing in certain places on clusters of certain sounds., which is more typical for intonationally emphasized and / or poetic texts with appealed or highlighted emotions. When using this property the first argument of the program call string does not matter (it is ignored by the program).
- '0y' is the first version of the rhythm-only analysis property (semi-empirical), based on the function of rhythm, that uses the durations of sounds that have been synthesized in the mmsyn6ukr-array software package. The rhythm function is inspired by antiquity poetry, where instead of stressed and unstressed syllables rhythmically short and long alternated; also musical destinies for which the main ones are two-part rhythm and three-part rhythm. The function is implemented in such a way as to make it as easy as possible to capture significant emissions subrhythms for two-syllable and three-syllable cases. Using the <first argument> you can change the ratio of these subproperties and, accordingly, to change the property itself.

- '02y' is a '0y'-like property that uses other durations of sounds synthesized by the r-glpk-phonetic-languages-ukrainian-durations. You can create other variations of sound durations using the capabilities of the package r-glpk-phonetic-languages-ukrainian-durations or in some other way.
- '03y' is a '02y'-like property that uses other durations of sounds synthesized by the r-glpk-phonetic-languages-ukrainian-durations package. You can create other variations of sound durations using the capabilities of the package r-glpk-phonetic-languages-ukrainian-durations or in some other way.
- '04y' is a '02y'-like property that uses other durations of sounds synthesized by the r-glpk-phonetic-languages-ukrainian-durations package. These sound durations are derived from data other than 0y, 02y and 03y, so be careful while mixing them in a multiple properties mode.
- 'y' - a property that calculates the properties of 'y0' and '0y' in a more efficient way than each of them alone, and then multiplies the received data. Gives higher values for lines with a more distant location of the words with the same sounds one from the others and those that are more rhythmic (from the point of view of the property '0y'). The use of the <first coefficient> internally affects only the sub-property '0y'.
- 'y2' is a property similar to 'y', but uses the variant with '02y' instead of the second sub-property (rhythmicity).
- 'y3' is a property similar to 'y', but uses the variant with '03y' instead of the second sub-property (rhythmicity).
- 'y4' is a property similar to 'y', but uses the variant with '04y' instead of the second sub-property (rhythmicity).
- 'yy' - a property that uses the property 'y0' and '0y', and instead of multiplying them, divides the result of the second by the result of the first. Maximized for texts with high rhythmicity (in terms of the '0y' property) and grouping of identical sounds into the groups that are closer to each other. The use of the <first coefficient> affects only the sub-property '0y'.
- 'yy2' - a property that uses the property 'y0' and '02y', and instead of multiplying them, divides the result of the second by the result of the first. Maximized for texts with high rhythmicity (in terms of the property '02y') and grouping of identical sounds in groups closer to each other. The use of the <first coefficient> internally affects only the sub-property '02y'.

- 'yy3' - a property that uses the property 'y0' and '03y', and instead of multiplying them, divides the result of the second by the result of the first. Maximized for texts with high rhythmicity (in terms of the property '03y') and grouping of identical sounds in groups closer to each other. The use of the <first coefficient> internally affects only the sub-property '03y'.
- 'yy4' - similar to 'yy' with the difference that instead of '0y' is used '04y'.

'Z' -line

- '0z'
- '02z'
- '03z'
- '04z'
- 'z'
- 'z2'
- 'z3'
- 'z4'
- 'zz'
- 'zz2'
- 'zz3'
- 'zz4' These properties are similar to the corresponding ones, where z is replaced by y. But they use more complex rhythmic functions derived from the module `Languages.Rhythmicity.Factor` from the package `phonetic-languages-rhythmicity`. Carefully use mixed properties in multiple properties mode, because they actually represent different approaches within the general method, so they can give in pairs less compatible results, but when used correctly they give an acceptable result. You may need a little practice, also more often use the `propertiesTextG` program.

Besides, while working with the following properties the concept of the polyrhythmicity as a source of rhythmicity is used.

- "w01" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one maximum value the most significantly influences the rhythmicity and another maximum one influences less. As a variant of the durations calculation function syllableDurationsD is used;
- "w02" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one maximum value the most significantly influences the rhythmicity and another maximum one influences less. As a variant of the durations calculation function syllableDurationsD2 is used;
- "w03" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one maximum value the most significantly influences the rhythmicity and another maximum one influences less. As a variant of the durations calculation function syllableDurationsD3 is used;
- "w04" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one maximum value the most significantly influences the rhythmicity and another maximum one influences less. As a variant of the durations calculation function syllableDurationsD4 is used;
- "w11" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the two maximum values the most significantly influence the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD is used;
- "w12" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the two maximum values the most significantly influence the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD2 is used;
- "w13" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the two maximum values the most significantly influence the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD3 is used;
- "w14" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the two maximum values the most significantly influence the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD4 is used;

- "w21" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and two maximum ones influence less. As a variant of the durations calculation function syllableDurationsD is used;
- "w22" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and two maximum ones influence less. As a variant of the durations calculation function syllableDurationsD2 is used;
- "w23" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and two maximum ones influence less. As a variant of the durations calculation function syllableDurationsD3 is used;
- "w24" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and two maximum ones influence less. As a variant of the durations calculation function syllableDurationsD4 is used;
- "w31" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD is used;
- "w32" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD2 is used;
- "w33" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD3 is used;
- "w34" – more complex rhythmic structure with some another way of determining the property value, it takes the string as the text with rhythmic groups of 4 syllables each where the one minimum value the most significantly influences the rhythmicity and another minimum one influences less. As a variant of the durations calculation function syllableDurationsD4 is used;



- "x01" – similarly to the "w01", but with more complex dependency for the less significant duration and probably less prognosable results;
- "x02" – similarly to the "w02", but with more complex dependency for the less significant duration and probably less prognosable results;
- "x03" – similarly to the "w03", but with more complex dependency for the less significant duration and probably less prognosable results;
- "x04" – similarly to the "w04", but with more complex dependency for the less significant duration and probably less prognosable results;

The following values are similar to the corresponding "w" with more complex dependency (as just described ones). Among them:

- "x11"
- "x12"
- "x13"
- "x14"
- "x21"
- "x22"
- "x23"
- "x24"
- "x31"
- "x32"
- "x33"

- "x34"

If this argument is as follows, then a polyrhythmic analysis of the text is used. More complex properties of the text are searched and checked using more comprehensive by structure properties. This is a research direction in the programs and library usage. Besides there is a possibility also to specify your own custom configuration using the 'c', 'C', 'N' modes. Therefore, the following is used:

- "u01" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 and even less highlighted 1 maximae, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u02" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 and even less highlighted 1 maximae, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u03" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 and even less highlighted 1 maximae, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u04" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 and even less highlighted 1 maximae, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u11" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 maximae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u12" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 maximae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u13" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 maximae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u14" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 maximae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u21" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;

- "u22" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u23" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u24" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u31" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u32" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u33" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u34" -> A polyrhythm with the most highlighted 1 maximum, less highlighted 2 minimae and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u41" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u42" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u43" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u44" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;

- "u51" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u52" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u53" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u54" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 maximum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u61" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u62" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u63" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;
- "u64" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 maximum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "u71" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD;
- "u72" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD2;
- "u73" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD3;

- "u74" -> A polyrhythm with the most highlighted 2 maximae, less highlighted 1 minimum and even less highlighted 1 minimum, the groups of 5 syllables. For syllable durations is used syllableDurationsD4;
- "v01" and other "v" – are analogous to the corresponding "u" lines with that difference that they use only increasing function variants for the rhythmicity estimation. This makes them the more straightforward ones.
- "s01" and other "s" – are analogous to the corresponding "u" lines with that difference that they group syllables into groups of 6, not 5.
- "t01" and other "t" lines – are analogous to the corresponding "s" lines with that difference that they only use increasing function variants for the rhythmicity estimation. This makes them the more straightforward ones.
- "S" line corresponds to the properties of the "s" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF2;
- "T" line corresponds to the properties of the "t" line with the difference that the following function is used: i. e. rhythmicityPoly-WeightedF20;
- "U" line corresponds to the properties of the "u" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF2;
- "V" line corresponds to the properties of the "v" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF20;
- "W" line corresponds to the properties of the "u" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF3;
- "X" line corresponds to the properties of the "v" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF30;
- "Y" line corresponds to the properties of the "s" line with the difference that the following function is used: i. e. rhythmicity-PolyWeightedF3;

- “Z” line corresponds to the properties of the “t” line with the difference that the following function is used: i. e. `rhythmicityPolyWeightedF30`;

The following property lines try to increase the significance of the text ending and decrease the significance of its beginning.

- “I” line corresponds to the properties of the “W” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF2`;
- “J” line corresponds to the properties of the “X” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF20`;
- “K” line corresponds to the properties of the “Y” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF2`;
- “L” line corresponds to the properties of the “Z” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF20`;
- “O” line corresponds to the properties of the “U” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF3`;
- “P” line corresponds to the properties of the “V” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF30`;
- “Q” line corresponds to the properties of the “S” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF3`;
- “R” line corresponds to the properties of the “T” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedEF30`;

Starting from the version 0.10.0.0 there are introduced also the following properties:

- “o” line corresponds to the properties of the “u” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedLF2`;

- “p” line corresponds to the properties of the “v” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF20;
- “q” line corresponds to the properties of the “s” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF2;
- “r” line corresponds to the properties of the “t” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF20;
- “k” line corresponds to the properties of the “u” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF3;
- “l” line corresponds to the properties of the “v” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF30;
- “m” line corresponds to the properties of the “s” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF3;
- “n” line corresponds to the properties of the “t” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLF30;

The following lines try to increase the importance of the line ending and to decrease the importance of the its beginning.

- “g” line corresponds to the properties of the “u” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLEF2;
- “h” line corresponds to the properties of the “v” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLEF20;
- “i” line corresponds to the properties of the “s” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLEF2;
- “j” line corresponds to the properties of the “t” line with the difference that the following weighted function is used: rhythmicityPolyWeightedLEF20;

- “b” line corresponds to the properties of the “u” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedLEF3`;
- “d” line corresponds to the properties of the “v” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedLEF30`;
- “e” line corresponds to the properties of the “s” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedLEF3`;
- “f” line corresponds to the properties of the “t” line with the difference that the following weighted function is used: `rhythmicityPolyWeightedLEF30`;

#### Custom configuration properties

If the property type starts with one of the “c”, “A”, “B”, “C”, “D”, “E”, “F”, “M”, “N” then the program tries to parse this property as an encoded configuration of the polyrhythmicity. A mode for the developers and researchers, there can be used even more complex polyrhythmic structures. For example, “c114+112=2” returns as a polyrhythm structure data P1 (Ch 1 1 4) (Rhythm 1 1 2) 2, that means that the 1 most highlighted maximum, and 1 less highlighted maximum is searched in the groups of 4 syllables using `syllableDurationsD2`; “ctttff7+112111=7\*3” returns as a polyrhythm structure data P2 (PolyCh [True,True,True,False,False] 7) (PolyRhythm [1,1,2,1,1]) 7 3, that means that the 1 most highlighted maximum, and 1 less highlighted maximum, and 2 even less highlighted maximae, and 1 even less highlighted minimum, and 1 even less highlighted minimum in the groups of 7 syllables using the `syllableDurationsD3` etc.

While using the new properties of the “A”, “B”, “C”, “D”, “E”, “F”, “M” and “N” lines there will be used ‘weighted’ functions (highly experimental ones, though hopefully effective) that try to take into account also the significance of the place of the line part e. g. internally there will be used such functions as `rhythmicityPolyWeightedLEF2`, `rhythmicityPolyWeightedEF2`, `rhythmicityPolyWeightedF2`, `rhythmicityPolyWeightedLF2`, `rhythmicityPolyWeightedLEF3`, `rhythmicityPolyWeightedLF3`, `rhythmicityPolyWeightedEF3` and `rhythmicityPolyWeightedF3` respectively, all from the module `Rhythmicity.PolyRhythm`.

Tip: If there are no results in the program output using multiple properties (an empty output), increase the number of groups in properties (for at least one) and / or add interval transforms to change the internal structure of certain properties.

For more convenience and easiness there is a table further with the partial classification of the properties.



| Properties   | Groups by N, N | Attention to | Linearity | Is there Infinity | Peculiarities              | Used in | Column 2 |
|--------------|----------------|--------------|-----------|-------------------|----------------------------|---------|----------|
| y0           |                | 0            | 0         | 0                 | Uniqueness periods         | 1-3     |          |
| 0{M}y, 0{M}z | 2   3          | 0            | 0         | 0                 | First argument             | 1-3     |          |
| y{M}, z{M}   | 2   3          | 0            | 0         | 0                 | First argument             | 1-3     |          |
| yy{M}, zz{M} | 2   3          | 0            | 0         | 0                 | First argument             | 1-3     |          |
| w, x         | 4              | 0            | 0         | 1                 | wx-argument                | 1-3     |          |
| u, v,        | 5              | 0            | 0         | 1                 |                            | 1-3     |          |
| s, t,        | 6              | 0            | 0         | 1                 |                            | 1-3     |          |
| S, T,        | 6              | 1            | 0         | 1                 |                            | 1-3     |          |
| U, V,        | 5              | 1            | 0         | 1                 |                            | 1-3     |          |
| W, X,        | 5              | 1            | 0         | 1                 |                            | 1-3     |          |
| Y, Z,        | 6              | 1            | 0         | 1                 |                            | 1-3     |          |
| l, j,        | 5              | -1           | 0         | 1                 |                            | 1-3     |          |
| K, L,        | 6              | -1           | 0         | 1                 |                            | 1-3     |          |
| O, P,        | 5              | -1           | 0         | 1                 |                            | 1-3     |          |
| Q, R,        | 6              | -1           | 0         | 1                 |                            | 1-3     |          |
| o, p,        | 5              | 1            | 1         | 0                 |                            | 1-3     |          |
| q, r,        | 6              | 1            | 1         | 0                 |                            | 1-3     |          |
| k, l,        | 5              | 1            | 1         | 0                 |                            | 1-3     |          |
| m, n,        | 6              | 1            | 1         | 0                 |                            | 1-3     |          |
| g, h,        | 5              | -1           | 1         | 0                 |                            | 1-3     |          |
| i, j,        | 6              | -1           | 1         | 0                 |                            | 1-3     |          |
| b, d,        | 5              | -1           | 1         | 0                 |                            | 1-3     |          |
| e, f,        | 6              | -1           | 1         | 0                 |                            | 1-3     |          |
| c            | Customized     |              |           |                   | Customized structure       | 1-3     |          |
| A            | Customized     | -1           | 1         | 0                 | Customized structure       | 1-3     |          |
| B            | Customized     | 1            | 0         | 1                 | Customized structure       | 1-3     |          |
| C            | Customized     | 1            | 0         | 1                 | Customized structure       | 1-3     |          |
| D            | Customized     | 1            | 1         | 0                 | Customized structure       | 1-3     |          |
| E            | Customized     | -1           | 1         | 0                 | Customized structure       | 1-3     |          |
| F            | Customized     | 1            | 1         | 0                 | Customized structure       | 1-3     |          |
| M            | Customized     | -1           | 0         | 1                 | Customized structure       | 1-3     |          |
| N            | Customized     | 1            | 0         | 1                 | Customized structure       | 1-3     |          |
| H...         |                |              |           |                   | Durations                  | 1-3     |          |
| G...         |                |              |           |                   | Other scheme               | 1       |          |
| a...         |                |              |           |                   | syllable-as-a-whole (SaaW) | 1       |          |

In a table in column «Attention to» value 0 means that the property has the same effect on processing data that stands in all complete groups (if any) or throughout the line, if any; the value of 1 means, that the program is more sensitive to the beginning of the line; the value -1 means that the program is more sensitive to the latest groups (if any), or simply before the end of the line (if there are no groups). Arguments 1 and 0 correspond to "yes" and "no" in the next two columns. Thus, for example, the string o, p, 5 1 1 0 1-3 means that for properties with the first letters in the designation o or p takes place as follows: they group the syllables into groups of 5 syllables, starting from the beginning of the line, all syllables that can not be grouped (at the end, a certain "excess"), do not count, the properties are more sensitive to the beginning of the string, are "linear" (such is the function that forms the nucleus of increasing or decreasing, relative to the movement along the line forward or backward), and accordingly, it cannot acquire Infinity values for more or less "real" strings; the properties can be used in all three programs:

1. lineVariantsG3
2. propertiesTextG3
3. rewritePoemG3

• If in the column 'Used in' there is some number then the corresponding program can use it.

# The first argument

For the first time the program can be used without this argument, or by entering in its place 1\_. You may want to in the future to deepen the analysis. Then you can also specify the first argument (it is the first in the list of command line arguments, not counting the group - if available - between arguments +a and -a) in the form number1\_number2, where number1 and number2 - decimal positive fractions or positive integer numbers, and one of them may be absent (then it is considered equal to 1). Example,

3.4\_2  
2.987\_0.7865  
0.0001\_  
etc.

The first number is a coefficient multiplied by the component of the property that is responsible for the rhythmicity of the two-syllables based verse system, and the second – for rhythmicity of three-syllables based verse system. Accordingly, a successful combination can emphasize, combine or eliminate, reduce the effect of rhythm for two-component or three-component based verse system.

In the absence of this argument, the programs behave as if it were equal to 1\_1.

I must say that this argument can significantly affect the results and can change them to almost the opposite. Consider the following example.

```
propertiesTextG3 20000_1 Shchaslyve_derevo.txt s 1 0 +d 56.csv H02z | distributionTextG s 1 +W
```

|      |   |                         |       |        |         |         |            |   |
|------|---|-------------------------|-------|--------|---------|---------|------------|---|
| 2226 | 5 | 2226                    | 51204 | 1.0000 | 23.0004 | 23.0004 | 0.08333195 | 4 |
| 1    |   | Яке воно щасливе дерево |       |        |         |         |            |   |

|       |        |         |            |             |             |            |   |
|-------|--------|---------|------------|-------------|-------------|------------|---|
| 0     | 1      | 204801  | Infinity   | Infinity    | 204801.0000 | 0.00000977 |   |
| 1     |        |         |            |             |             |            |   |
| 225   | 2447   | 20144   | 10.8765    | 89.5289     | 8.2314      | 0.24028889 | 3 |
| 1     |        |         |            |             |             |            |   |
| 11251 | 20009  | 20009   | 1.7784     | 1.7784      | 1.0000      | 1.28016635 | 3 |
| 5     |        |         |            |             |             |            |   |
| 881   | 1025   | 20081   | 1.1635     | 22.7934     | 19.5912     | 0.09779601 | 3 |
| 1     |        |         |            |             |             |            |   |
| 144   | 20225  | 51344   | 140.4514   | 356.5556    | 2.5386      | 0.78561995 | 4 |
| 2     |        |         |            |             |             |            |   |
| 1     | 51204  | 204801  | 51204.0000 | 204801.0000 | 3.9997      | 0.50003418 |   |
| 2     |        |         |            |             |             |            |   |
| 4     | 80016  | 720004  | 20004.0000 | 180001.0000 | 8.9983      | 0.22226420 |   |
| 1     |        |         |            |             |             |            |   |
| 4     | 5016   | 204804  | 1254.0000  | 51201.0000  | 40.8301     | 0.04898246 | 3 |
| 1     |        |         |            |             |             |            |   |
| 7200  | 20001  | 460804  | 2.7779     | 64.0006     | 23.0390     | 0.08547363 | 4 |
| 1     |        |         |            |             |             |            |   |
| 800   | 3200   | 13889   | 4.0000     | 17.3611     | 4.3403      | 0.43570348 | 4 |
| 1     |        |         |            |             |             |            |   |
| 36    | 36     | 51236   | 1.0000     | 1423.2222   | 1423.2222   | 0.00140428 | 3 |
| 1     |        |         |            |             |             |            |   |
| 2226  | 20004  | 51201   | 8.9856     | 22.9991     | 2.5595      | 0.74883174 | 3 |
| 2     |        |         |            |             |             |            |   |
| 16    | 1266   | 1280016 | 79.1250    | 80001.0000  | 1011.0711   | 0.00197808 |   |
| 1     |        |         |            |             |             |            |   |
| 20004 | 80004  | 460804  | 3.9994     | 23.0356     | 5.7598      | 0.33278980 | 3 |
| 1     |        |         |            |             |             |            |   |
| 45000 | 204801 | 204804  | 4.5511     | 4.5512      | 1.0000      | 1.63969352 | 3 |

|       |                                     |        |            |             |            |            |  |   |
|-------|-------------------------------------|--------|------------|-------------|------------|------------|--|---|
| 5     | дзвінким відспіву птахів            |        |            |             |            |            |  |   |
| 20004 | 51204                               | 204804 | 2.5597     | 10.2382     | 3.9998     | 0.45553539 |  | 3 |
| 1     | Уранці ввечері чивдень              |        |            |             |            |            |  |   |
| 36    | 5144                                | 819344 | 142.8889   | 22759.5556  | 159.2815   | 0.01255583 |  |   |
| 1     | постійно бачиш його квітним         |        |            |             |            |            |  |   |
| 1     | 20004                               | 204804 | 20004.0000 | 204804.0000 | 10.2382    | 0.19534679 |  |   |
| 1     | Підійдеш ближче ось воно            |        |            |             |            |            |  |   |
| 1     | 1254                                | 45009  | 1254.0000  | 45009.0000  | 35.8923    | 0.05572095 |  |   |
| 1     | Відходиш далі хочеш бути знов       |        |            |             |            |            |  |   |
| 4     | 20016                               | 35572  | 5004.0000  | 8892.8889   | 1.7772     | 1.12526704 |  | 3 |
| 3     | Авзимку чищасливе дерево            |        |            |             |            |            |  |   |
| 36    | 5009                                | 80036  | 139.1389   | 2223.2222   | 15.9784    | 0.12511240 |  | 4 |
| 81    | 20081                               | 80225  | 247.9136   | 990.4321    | 3.9951     | 0.50011207 |  | 5 |
| 2     | Асніг тепло оберігає тарадує людей  |        |            |             |            |            |  |   |
| 9     | 25                                  | 720009 | 2.7778     | 80001.0000  | 28800.3600 | 0.00006944 |  | 4 |
| 1     | свята Різдвяна радість огортає      |        |            |             |            |            |  |   |
| 1     | 16                                  | 460804 | 16.0000    | 460804.0000 | 28800.2500 | 0.00006944 |  |   |
| 1     | Длядерева затих той теплий рух      |        |            |             |            |            |  |   |
| 1     | 3674                                | 35560  | 3674.4694  | 35559.5556  | 9.6775     | 0.20665984 |  | 5 |
| 1     | Тасили додає йому той зріст         |        |            |             |            |            |  |   |
| 36    | 36                                  | 51236  | 1.0000     | 1423.2222   | 1423.2222  | 0.00140428 |  | 3 |
| 1     | ігнуцкість іміцне здоров'я          |        |            |             |            |            |  |   |
| 0     | 45009                               | 460809 | Infinity   | Infinity    | 10.2382    | 0.19534775 |  | 6 |
| 1     | Цепори року йдерево просте Ами люди |        |            |             |            |            |  |   |
| 2226  | 51236                               | 51236  | 23.0148    | 23.0148     | 1.0000     | 1.91671793 |  | 3 |
| 5     | Багатші ірухливі іповсюди           |        |            |             |            |            |  |   |
| 5016  | 204804                              | 720004 | 40.8301    | 143.5415    | 3.5156     | 0.56496097 |  |   |
| 2     | Ущасті ми активні                   |        |            |             |            |            |  |   |
| 1254  | 1254                                | 8893   | 1.0000     | 7.0916      | 7.0916     | 0.24716936 |  | 2 |

142

1 айакнійдемо донього  
поступово  
людяно

| 1              | 2 | 3      | 4     | 5 |       |       |
|----------------|---|--------|-------|---|-------|-------|
| 22             | 5 | 1      | 0     | 3 |       |       |
| 70.97%         |   | 16.13% | 3.23% |   | 0.00% | 9.68% |
| 0.3905+-0.4959 |   | 0      | 31    |   |       |       |

| 2 | 3  | 4  | 5 | 6 | 7 |
|---|----|----|---|---|---|
| 1 | 14 | 10 | 5 | 1 | 0 |

\*\*\*\*\*

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | . | . | . | . |
| 7 | 3 | 1 | . | 3 |
| 9 | 1 | . | . | . |
| 4 | 1 | . | . | . |
| 1 | . | . | . | . |
| . | . | . | . | . |

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 7 | 3 | 1 | 0 | 3 |
| 9 | 1 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

propertiesTextG3 1\_20000 Shchaslyve\_derevo.txt s 1 0 +d 56.csv H02z | distributionTextG s 1 +W

5

|       |       |       |        |        |        |            |   |
|-------|-------|-------|--------|--------|--------|------------|---|
| 80000 | 80000 | 80003 | 1.0000 | 1.0000 | 1.0000 | 0.99998469 | 4 |
|-------|-------|-------|--------|--------|--------|------------|---|

|        |                               |         |            |              |             |            |  |   |
|--------|-------------------------------|---------|------------|--------------|-------------|------------|--|---|
| 1      | Яке воно щасливе дерево       |         |            |              |             |            |  |   |
| 0      | 20000                         | 80003   | Infinity   | Infinity     | 4.0001      | 0.49998400 |  | 5 |
| 2      | Ти можеш бачити цвіте воно    |         |            |              |             |            |  |   |
| 180001 | 4500000                       | 4500000 | 24.9999    | 24.9999      | 1.0000      | 1.92307669 |  |   |
| 5      | Закриєш очі плодоносить       |         |            |              |             |            |  |   |
| 20001  | 180001                        | 180001  | 8.9998     | 8.9998       | 1.0000      | 1.79999594 |  | 3 |
| 5      | Радітимеш побачивши його      |         |            |              |             |            |  |   |
| 720000 | 4500000                       | 4500001 | 6.2500     | 6.2500       | 1.0000      | 1.72413762 |  |   |
| 5      | іквітним іводночас плідним    |         |            |              |             |            |  |   |
| 720001 | 4500001                       | 6480002 | 6.2500     | 9.0000       | 1.4400      | 1.24999987 |  |   |
| 4      | Анавесні колищеквіту там нема |         |            |              |             |            |  |   |
| 20000  | 80003                         | 80003   | 4.0001     | 4.0001       | 1.0000      | 1.60001024 |  | 3 |
| 5      | Чийе щасливим дерево          |         |            |              |             |            |  |   |
| 80000  | 320004                        | 320004  | 4.0000     | 4.0000       | 1.0000      | 1.60000400 |  | 4 |
| 5      | Напевне Унадії втій порі      |         |            |              |             |            |  |   |
| 80000  | 320000                        | 320000  | 4.0000     | 4.0000       | 1.0000      | 1.60000025 |  | 3 |
| 5      | щосонце гріє Івранці          |         |            |              |             |            |  |   |
| 0      | 20001                         | 320001  | 55558.3333 | 888891.6667  | 15.9993     | 0.12500572 |  |   |
| 1      | сяють теплі йпрохолодні роси  |         |            |              |             |            |  |   |
| 0      | 0                             | 80000   | 4.0000     | 2000009.0000 | 500002.2500 | 0.00000400 |  |   |
| 1      | Приходиш ти донього навесні   |         |            |              |             |            |  |   |
| 720000 | 720000                        | 720003  | 1.0000     | 1.0000       | 1.0000      | 0.99999822 |  |   |
| 1      | ібачиш дерево щасливим        |         |            |              |             |            |  |   |
| 20003  | 80001                         | 80001   | 3.9995     | 3.9995       | 1.0000      | 1.59996304 |  | 3 |
| 5      | Прийдеш улітку восени         |         |            |              |             |            |  |   |
| 80000  | 320000                        | 320064  | 4.0000     | 4.0008       | 1.0002      | 1.59974371 |  | 4 |
| 5      | ізноу зустрічаєш його милим   |         |            |              |             |            |  |   |
| 80001  | 80004                         | 320003  | 1.0000     | 4.0000       | 3.9998      | 0.40001644 |  | 3 |
| 1      | усміхненим красивим і         |         |            |              |             |            |  |   |

|        |                                     |         |             |              |         |            |   |
|--------|-------------------------------------|---------|-------------|--------------|---------|------------|---|
| 2      | 20010                               | 80010   | 8893.4400   | 35560.1067   | 3.9985  | 0.50017791 | 3 |
| 2      | дзвінким відспіву птахів            |         |             |              |         |            |   |
| 80001  | 80003                               | 180003  | 1.0000      | 2.2500       | 2.2500  | 0.61539588 | 3 |
| 1      | Уранці ввечері чивдень              |         |             |              |         |            |   |
| 720000 | 2880000                             | 2880041 | 4.0000      | 4.0001       | 1.0000  | 1.59998193 |   |
| 5      | постійно бачиш його квітним         |         |             |              |         |            |   |
| 20000  | 80001                               | 180003  | 4.0000      | 9.0001       | 2.2500  | 0.79999976 | 4 |
| 2      | Підійдеш ближче ось воно            |         |             |              |         |            |   |
| 0      | 80000                               | 320001  | 720000.5625 | 2880009.0000 | 4.0000  | 0.49999865 |   |
| 2      | Відходиш далі хочеш бути знов       |         |             |              |         |            |   |
| 80000  | 320001                              | 320002  | 4.0000      | 4.0000       | 1.0000  | 1.59999789 | 3 |
| 5      | Авзимку чищасливе дерево            |         |             |              |         |            |   |
| 0      | 180000                              | 720004  | 720001.0000 | 2880016.0000 | 4.0000  | 0.49999774 |   |
| 2      | Так боземля його тримає             |         |             |              |         |            |   |
| 720000 | 1620001                             | 4500004 | 2.2500      | 6.2500       | 2.7778  | 0.62068956 |   |
| 2      | Асніг тепло оберігає тарадує людей  |         |             |              |         |            |   |
| 80002  | 500000                              | 500002  | 6.2498      | 6.2499       | 1.0000  | 1.72412455 | 4 |
| 5      | свята Різдяна радість огортає       |         |             |              |         |            |   |
| 20000  | 320000                              | 320004  | 16.0000     | 16.0002      | 1.0000  | 1.88233080 | 5 |
| 5      | Длядерева затих той теплий рух      |         |             |              |         |            |   |
| 0      | 20000                               | 320001  | 125001.1480 | 2000006.2500 | 15.9999 | 0.12500069 |   |
| 1      | Тасили додає йому той зріст         |         |             |              |         |            |   |
| 720000 | 720000                              | 720003  | 1.0000      | 1.0000       | 1.0000  | 0.99999822 | 3 |
| 1      | ігнучкість імічне здоров'я          |         |             |              |         |            |   |
| 0      | 180002                              | 500002  | Infinity    | Infinity     | 2.7778  | 0.72000644 | 6 |
| 2      | Цепори року йдерево просте Ами люди |         |             |              |         |            |   |
| 80000  | 720003                              | 720003  | 9.0000      | 9.0000       | 1.0000  | 1.80000039 | 3 |
| 5      | Багатші ірухливі іповсюди           |         |             |              |         |            |   |
| 80010  | 80010                               | 320004  | 1.0000      | 3.9995       | 3.9995  | 0.40003696 | 3 |



| 1                  | 2  | 3      | 4     | 5  | 6     | 7      |
|--------------------|----|--------|-------|----|-------|--------|
| 1                  | 2  | 3      | 4     | 5  | 6     | 7      |
| 10                 | 7  | 0      | 1     | 13 |       |        |
| 32.26%             |    | 22.58% | 0.00% |    | 3.23% | 41.94% |
| 1.0681+<br>-0.6072 |    | 0      |       | 31 |       |        |
| 1                  | 3  | 4      | 5     | 6  | 7     |        |
| 1                  | 14 | 10     | 5     | 1  | 0     |        |
| *****              |    |        |       |    |       |        |
| 1                  | .  | .      | .     | .  |       |        |
| 5                  | 1  | .      | .     | 8  |       |        |
| 3                  | 2  | .      | 1     | 4  |       |        |
| 1                  | 3  | .      | .     | 1  |       |        |
| .                  | 1  | .      | .     | .  |       |        |
| .                  | .  | .      | .     | .  |       |        |
| ~~~~~              |    |        |       |    |       |        |
| 1                  | 0  | 0      | 0     | 0  |       |        |
| 5                  | 1  | 0      | 0     | 8  |       |        |
| 3                  | 2  | 0      | 1     | 4  |       |        |
| 1                  | 3  | 0      | 0     | 1  |       |        |
| 0                  | 1  | 0      | 0     | 0  |       |        |
| 0                  | 0  | 0      | 0     | 0  |       |        |

In the first case, the first argument significantly strengthens the analysis of the 2-syllable component, in the second - for the 3-syllable one. If you compare the statistics highlighted in red (mean and standard sample square deviation) in both cases, you will

see that at the text level, the amplification of 2-component components gives much less than 1 result, which means that the data is strongly shifted to the left, to smaller values, and in the second case gives even more than 1 result, which means that the data is more offset to the right, to larger values. But this does not mean that the text is closer to poems with three-syllable verse size. Many lines are classic lines with a two-syllable size (for example, iambic). in particular: "свѣта Рїздвяна радїсть огортає" (highlighted in blue), but in terms of the duration of the syllables, it is much more suitable for three-syllables foot. This is also an example of the difference between syllabic and tonic verse.

# Constraints

When you run the lineVariantsG3 program you can specify as command arguments constraints string. They allow to reduce the number of calculations, to consider only certain options (for example, with a certain definite order of some words, etc.) that allows you to actually expand the program capabilities. These limitations are encoded as command arguments line between two special notation 'brackets' '+a' and '-a'. They form a group of arguments that can stand anywhere in the input line data. Depending on these arguments, the program asks or does not ask an additional question for verification and confirmation (that is called double check).

There are 6 types of constraints, they can be arbitrarily combined, but with respect to boundaries for each of them.

The figure shows that all types implemented with one argument that is the same for all of them – the number of words (or their combinations) in a line. The user, having started the program, can no longer adjust during its work this amount, but it is important for limitations in general. None of the digital characters in the constraints should be greater than this number, also this number is not more than 6 and not less than 0. Also a necessary condition is that no numeric characters within one encoded constraint cannot be repeated twice. For example, the following constraints are obviously not valid: Q2235 (repetition of digits), E2 (digital characters where they do not exist), T247 (7 is greater than 6), F0 (one character instead of the required two), A37523 (7 is greater than 6), B5 (one symbol, and there must be other(s) one(s)). Incorrectly set constraints will either not affect the result (although it will be expected otherwise), or will cause runtime exception and program shutdown. Since the result of their application is not simple, so the program at its work displays a line to which the entered constraints will be applied with an additional question, whether all data are entered correctly.

The types of constraints and their values are given in more detail in the table.

- Constraint E – Without entering additional digital characters – Corresponds to the absence of additional constraints, so does not affects the end result.

- Constraint Q – 4 pairs of unequal digits in range from 0 to the number of words or their concatenations minus 1 – Numbers are indices of 4 words or their concatenations, the mutual order of which during the permutations will be saved as follows. Also, if these words are the same (excluding uppercase and lowercase letters), then it is a convenient way to reduce the amount of data to be analyzed.
- Constraint T – 3 pairs of unequal digits in range from 0 to the number of words or their concatenations minus 1 – Numbers are indices of 3 words or their concatenations, the mutual order of which during the permutations will be saved as follows. Also, if these words are the same (excluding uppercase and lowercase letters), then it is a convenient way to reduce the amount of data to be analyzed.
- Constraint F – 2 pairs of unequal digits in range from 0 to the number of words or their combinations minus 1 – Numbers are indices of 2 words or their concatenations, the mutual order of which during the permutations will be saved as follows. Also, if these words are the same (excluding uppercase and lowercase letters), then it is a convenient way to reduce the amount of data to be analyzed.
- Constraint A – 1 digit and a few more in pairs unequal numbers (all among themselves unequal) to the right of it within – The first digit is the index of the element relative to which the placement of all other elements is determined (words or their combinations); all other numbers on the right are indices of the elements that should stand in the resulting permutations to the RIGHT of the element with from the element with the index equal to the first digit.
- Constraint B – 1 digit and a few more in pairs unequal numbers (all among themselves unequal) to the right of it within – The first digit is the index of the element relative to which the placement of all other elements is determined (words or their combinations); all other numbers on the right are indices of the elements that should stand in the resulting permutations to the LEFT of the element with from the element with the index equal to the first digit.
- Constraint P (fixed Point) (introduced starting with version 0.18.0.0) – 1 or several more pairwise unequal digits (all are unequal among themselves) in the range from 0 to the number of words or their combinations minus 1 - each of them (except 0) – means the serial number of the word, which, when reshuffling, will remain in its place (the countdown begins with 1 from left to right), the number 0 is used only for the case of incomplete sets of permutations and corresponds to the situation when the last possible word, the 10th one, should remain in its place. In the case of full permutations set to leave the last word on its place, please, use instead the +b (+bl) command line parameter.

# Parallel execution of programs

Typically, all packet programs run on a single processor core. In this case, for all programs under consideration, there is an opportunity to enable multi-core operation - parallel computing. To do this, the command line arguments must include the following:

```
+RTS -N -RTS
```

Their placement does not affect the order and value of other command line arguments, and there may be other RTS entries parameters. For more information on these parameters, see the documentation [1].

We can only recommend these settings for the propertiesTextG3 program. For other programs, they are not recommended, although you can use them (they will simply increase the use of resources).

## Acknowledgements

The author would like to thank the authors of the work Provably Correct, Asymptotically Efficient, Higher-Order Reverse-Mode Automatic Differentiation Faustyna Krawiec, Simon Peyton-Jones, Neel Krishnaswami, Tom Ellis, Richard A. Eisenberg та Andrew Fitzgibbon for the idea of the optimization and Mikolaj Konarski, who drew the author's attention to this article. The author also thanks his friends, whom he wanted to please with this research. Moreover, the author is acknowledged to everyone who sincerely and kindly has facilitated and promoted the research in any way.

# Bibliography

- [1] Glasgow haskell compiler user's guide. 7.5. using smp parallelism. [Електронний ресурс]. Режим доступу: [https://downloads.haskell.org/ghc/latest/docs/users\\_guide.pdf](https://downloads.haskell.org/ghc/latest/docs/users_guide.pdf). Перевірено 10 листопада 2020 р.
- [2] Shlomo Alexander and Raymond Orbach. Density of states on fractals:«fractons». *Journal de Physique Lettres*, 43(17):625–631, 1982.
- [3] Jan Andres, Jiří Fišer, Grzegorz Gabor, and Krzysztof Leśniak. Multivalued fractals. *Chaos, Solitons & Fractals*, 24(3):665–700, 2005.
- [4] Jan Andres and Miposlav Rypka. Multivalued fractals and hyperfractals. *International Journal of Bifurcation and Chaos*, 22(01):1250009, 2012.
- [5] Corine Astésano. *Rythme et accentuation en français: invariance et variabilité stylistique*. Editions L'Harmattan, 2001.
- [6] David Avnir, Ofer Biham, Daniel Lidar, and Ofer Malcai. Is the geometry of nature fractal? *Science*, 279(5347):39–40, 1998.
- [7] Michael F Barnsley, John E Hutchinson, and Örjan Stenflo. V-variable fractals: fractals with partial self similarity. *Advances in Mathematics*, 218(6):2051–2088, 2008.
- [8] Christopher J Bishop and Yuval Peres. *Fractals in probability and analysis*, volume 162. Cambridge University Press, 2017.
- [9] Marilyn G Boltz. Tempo discrimination of musical patterns: Effects due to pitch and rhythmic structure. *Perception & Psychophysics*, 60(8):1357–1373, 1998.
- [10] Nia Cason, Corine Astésano, and Daniele Schön. Bridging music and speech rhythm: Rhythmic priming and audio–motor training affect speech perception. *Acta psychologica*, 155:43–50, 2015.

- [11] Andrés E Coca, Gerard O Tost, and Zhao Liang. Controlling chaotic melodies. *Proc. Encuentro Nacional de Investigación en Posgrados (ENIP)*, 2009.
- [12] G.W. Cooper, G. Cooper, L.B.A. MEYER, and L.B. Meyer. *The Rhythmic Structure of Music*. Phoenix books. University of Chicago Press, 1963.
- [13] Joseph R Daniele and Aniruddh D Patel. An empirical study of historical patterns in musical rhythm: Analysis of german & italian classical music using the npvi equation. *Music Perception: An Interdisciplinary Journal*, 31(1):10–18, 2013.
- [14] Norberto Degara, Matthew EP Davies, Antonio Pena, and Mark D Plumbley. Onset event decoding exploiting the rhythmic structure of polyphonic music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1228–1239, 2011.
- [15] Norberto Degara, Antonio Pena, Matthew EP Davies, and Mark D Plumbley. Note onset detection using rhythmic structure. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5526–5529. IEEE, 2010.
- [16] Carolyn Drake and Caroline Palmer. Accent structures in music performance. *Music perception*, 10(3):343–378, 1993.
- [17] Antonio Galves, Jesus Garcia, Denise Duarte, and Charlotte Galves. Sonority as a basis for rhythmic class discrimination. In *Speech Prosody 2002, International Conference, 2002*.
- [18] EJ Garba. Music programming–rule–based randomization of melodic patterns. 2008.
- [19] Alex Goldsmith. Synthesising music: exploiting self-similarity using modular forms.
- [20] H. S. Gross. "prosody." *Encyclopedia Britannica*. <https://www.britannica.com/art/prosody>.
- [21] Erin E Hannon. Perceiving speech rhythm in music: Listeners classify instrumental songs according to language of origin. *Cognition*, 111(3):403–409, 2009.
- [22] Andre Holzapfel and Yannis Stylianou. Rhythmic similarity of music based on dynamic periodicity warping. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2217–2220. IEEE, 2008.
- [23] David Huron and Matthew Royal. What is melodic accent? converging evidence from musical practice. *Music Perception*, 13(4):489–516, 1996.

- [24] John E Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981.
- [25] John R Iversen, Aniruddh D Patel, and Kengo Ohgushi. Perception of rhythmic grouping depends on auditory experience. *The Journal of the Acoustical Society of America*, 124(4):2263–2271, 2008.
- [26] Vijay Iyer, Jeff Bilmes, Matt Wright, and David Wessel. A novel representation for rhythmic structure. In *Proceedings of the 23rd International Computer Music Conference*, pages 97–100. Citeseer, 1997.
- [27] Jun Kigami. *Analysis on fractals*. Number 143. Cambridge University Press, 2001.
- [28] Cyrille Magne, Mitsuko Aramaki, Corine Astesano, Reyna Leigh Gordon, Sølvi Ystad, Snorre Farner, Richard Kronland-Martinet, and Mireille Besson. Comparison of rhythmic processing in language and music: An interdisciplinary approach. *Journal of Music and Meaning*, 3, 2005.
- [29] Benoit B Mandelbrot and Benoit B Mandelbrot. *The fractal geometry of nature*, volume 1. WH freeman New York, 1982.
- [30] Kevin Merges. Fractals and art. 2005.
- [31] Ihor Nabytovych. ФРАКТАЛИ ТА ФРАКТАЛЬНІ СТРУКТУРИ У ХУДОЖНЬОМУ ТЕКСТІ (на прикладі прози Л. Керрола, КС Льюїса та ХЛ Борхеса). *ВІСНИК ЛЬВІВСЬКОГО УНІВЕРСИТЕТУ. Серія іноземні мови*, (18).
- [32] Katie Overy. Dyslexia and music: From timing deficits to musical intervention. *Annals of the New York academy of sciences*, 999(1):497–505, 2003.
- [33] Narayan Partap and Renu Chugh. Fixed point iterative techniques—an application to fractals. *International Journal of Research in Mathematics & Computation*, 4(1):1–7, 2016.
- [34] Aniruddh D Patel. Rhythm in language and music: parallels and differences. *Annals of the New York Academy of Sciences*, 999(1):140–143, 2003.
- [35] Aniruddh D Patel and Joseph R Daniele. An empirical comparison of rhythm in language and music. *Cognition*, 87(1):B35–B45, 2003.
- [36] Aniruddh D Patel and Joseph R Daniele. Stress-timed vs. syllable-timed music? a comment on huron and ollen (2003). *Music Perception*, 21(2):273–276, 2003.



- [37] Rohit Sunkam Ramanujam and Bill Lin. Randomized partially-minimal routing on three-dimensional mesh networks. *IEEE Computer Architecture Letters*, 7(2):37–40, 2008.
- [38] Franck Ramus. Acoustic correlates of linguistic rhythm: Perspectives. 2002.
- [39] Dietmar Saupe. Algorithms for random fractals. In *The science of fractal images*, pages 71–136. Springer, 1988.
- [40] Xi Shao, Changsheng Xu, and Mohan S Kankanhalli. Unsupervised classification of music genre using hidden markov model. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 3, pages 2023–2026. IEEE, 2004.
- [41] Joyce L Shields, Astrid McHugh, and James G Martin. Reaction time to phoneme targets as a function of rhythmic cues in continuous speech. *Journal of Experimental Psychology*, 102(2):250, 1974.
- [42] Robert S Strichartz. Fractals in the large. *Canadian Journal of Mathematics*, 50(3):638–657, 1998.
- [43] Neil P McAngus Todd. Segmentation and stress in the rhythmic structure of music and speech: A wavelet model. *The Journal of the Acoustical Society of America*, 93(4):2363–2363, 1993.
- [44] Wiebke Trost and Patrik Vuilleumier. Rhythmic entrainment as a mechanism for emotion induction by music: a neurophysiological perspective. *The emotional power of music: Multidisciplinary perspectives on musical arousal, expression, and social control*, pages 213–225, 2013.
- [45] Giuseppe Vitiello. The brain is like an orchestra. better yet, it is like a jazz combo, which doesn't need a conductor. *Chaos*, 11(1):2017, 2017.
- [46] Peter Vuust, Leif Ostergaard, Karen Johanne Pallesen, Christopher Bailey, and Andreas Roepstorff. Predictive coding of music-brain responses to rhythmic incongruity. *cortex*, 45(1):80–92, 2009.
- [47] Huang Wan-Li. The extremity laws of hydro-thermodynamics. *Applied Mathematics and Mechanics*, 4(4):501–510, 1983.
- [48] Kerri Welch. *A fractal topology of time: Implications for consciousness and cosmology*. California Institute of Integral Studies, 2010.

- [49] Wolfgang Wildgen. Chaos, fractals and dissipative structures in language. or the end of linguistic structuralism. *Gabriel Altmann und Walter A. Koch (Hg.), Systems. New Paradigms for the Human Sciences, de Gruyter, Berlin*, pages 596–620, 1998.
- [50] Marcel Zentner and Tuomas Eerola. Rhythmic engagement with music in infancy. *Proceedings of the National Academy of Sciences*, 107(13):5768–5773, 2010.
- [51] Oleksandr Zhabenko. dobutokO-poetry. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/dobutokO-poetry-0.8.1.0>. Перевірено 09 листопада 2020 р.
- [52] Oleksandr Zhabenko. phonetic-languages-rhythmicity. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/phonetic-languages-rhythmicity>. Перевірено 24 серпня 2020 р.
- [53] Oleksandr Zhabenko. phonetic-languages-simplified-generalized-examples-array. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/phonetic-languages-simplified-generalized-examples-array>. Перевірено 23 жовтня 2021 р.
- [54] П Івончак. Український силабо-тонічний вірш 50-х років XIX століття. *Науковий вісник Чернівецького національного університету. Слов'янська філологія*, (585-586):80–84, 2012.
- [55] Смаглий Г. А. Теорія музики : Підруч. для навч. закл. освіти, культури і мистецтв., 2013.
- [56] Під ред. Анатолія Волкова. Адекватний переклад // Лексикон загального та порівняльного літературознавства., 2001.
- [57] Під ред. Анатолія Волкова. Айрен // Лексикон загального та порівняльного літературознавства., 2001.
- [58] Під ред. Анатолія Волкова. Александрійський вірш // Лексикон загального та порівняльного літературознавства., 2001.
- [59] Під ред. Анатолія Волкова. Алкеєва строфа // Лексикон загального та порівняльного літературознавства., 2001.
- [60] Під ред. Анатолія Волкова. Алкманова або Архілохова строфа // Лексикон загального та порівняльного літературознавства., 2001.
- [61] Під ред. Анатолія Волкова. Античні розміри // Лексикон загального та порівняльного літературознавства., 2001.

- [62] Під ред. Анатолія Волкова. Античні строфи // Лексикон загального та порівняльного літературознавства., 2001.
- [63] Під ред. Анатолія Волкова. Аруз, або Аруд // Лексикон загального та порівняльного літературознавства., 2001.
- [64] Під ред. Анатолія Волкова. Асклепіадова строфа // Лексикон загального та порівняльного літературознавства., 2001.
- [65] Під ред. Анатолія Волкова. Баяті // Лексикон загального та порівняльного літературознавства., 2001.
- [66] Під ред. Анатолія Волкова. Буриме // Лексикон загального та порівняльного літературознавства., 2001.
- [67] Під ред. Анатолія Волкова. Варіативність // Лексикон загального та порівняльного літературознавства., 2001.
- [68] Під ред. Анатолія Волкова. Верлібр // Лексикон загального та порівняльного літературознавства., 2001.
- [69] МИКОЛА ВАСИЛЬОВИЧ Гуцуляк. *Українське віршування 30-80-х рр. XVII ст.* PhD thesis, «Теорія літератури». Чернівці: ЧНУ ім. Юрія Федьковича, 2017, 2017.
- [70] Наталія Костенко. ЕЛЕМЕНТИ ТОНІЧНОГО І СИЛАБІЧНОГО ВІРШУВАННЯ В ДУМОВОМУ ВІРШІ ТГ ШЕВЧЕНКА. *НАШ УКРАЇНСЬКИЙ ДІМ*, page 38.
- [71] АГ Кошовий and ГІ Кошовий. Одновимірні самоподібні фрактали та їх використання у моделюванні. 2011.
- [72] Оксана Валентинівна Кудряшова. *Functional poetics*. 2016.
- [73] О. В. Лазер-Паньків та ін. Л. Л. Звонська, Н. В. Корольова. Ямбічна строфа // *Енциклопедичний словник класичних мов.*, 2017.
- [74] ОВ Любімова. Відтворення античних розмірів в українській поезії 80-х–90-х років XIX століття. *Науковий вісник Чернівецького університету. Романо-слов'янський дискурс*, (565):190–193, 2011.
- [75] К Паладян. Початки силабо-тонічної версифікації в румунській літературі. *Питання літературознавства*, (81):164–172, 2010.
- [76] ВД Шарко. Актуальні проблеми природничо-математичної освіти в середній і вищій школі.
- [77] Тарас Шевченко. *Садок вишневий коло хати*. Strelbytskyy Multimedia Publishing, 2018.