Automatic cryptoanalysis of the monoalphabetical substitution as a method of the system approach in the algorithm development thinking

Stepan Hubalovsky, Michal Musilek

***Abstract*—**The system approach is one of the education methods which can be widely applied in any subjects. The authors have found this method as suitable for training in theoretical knowledge of algorithm development and programming. Cryptoanalysis has been chosen for the presentation of the system approach in this paper. The huge advantage of cryptoanalysis, which is a part of the curriculum in computer science, is that it enables the building of the system approach as well as interdisciplinary relations between such subjects as mother tongue, foreign languages, mathematics, history and geography. The use of the algorithm development and programming in the cryptoanalysis of the monoalphabetical substitution cipher is specifically presented in the paper.

***Keywords*—**Algorithm development, cryptoanalysis, interdisciplinary learning, monoalphabetical substitution cipher, system approach.

I. Introduction

THE algorithm development and programming belong to the main terms used in the computer science. Algorithms are encountered in all practical activities without being realized. An algorithm generally involves defining the rules and giving the sequences of steps necessary for any activities.

In a more narrow sense, an algorithm can be defined [1] as "a precise rule defining the process leading from mutable input data up to the required results". Such regulations are composed of various steps, stages, which are written in a particular order. A number of steps has to be finite. An algorithm can be understood as an activity which consists of problem analysis, design phases, a method of algorithm solution and finally of compilation of the relevant algorithmic language.

The most common examples of algorithms in everyday life are various descriptions and giving directions. The ability to create algorithms develops logical thinking skills and imagination and is an inseparable part of study skills of prospective and undergraduate teachers specializing in "Informatics" at the Faculty of Science.

II. The System Approach to the Algorithm Development and Programming

Teaching of the algorithm development as well as programming is usually carried out by teachers who also teach or are close to mathematics. The issue of the algorithm development is explained through mathematical tasks, which can be clearly described, defined and developed by algorithm. Altogether, the exercises are based on a routine rewriting of mathematical equations and formulas using algorithms and practicing the standard algorithm, regardless of their complexity and system integration to the whole entity [5], when the exercises used are "artificial", inconsistent and disconnected from reality.

In recent years the structure of teaching of the algorithm development at the University of Hradec Kralove has significantly moved towards the use of the system approach to teaching algorithms and programming. This shift can be understood as important progress for students - prospective teachers - who have to be able to apply new methods in their work and disseminate these new methods among secondary school students. Benefits of the new method in teaching and learning computer science is also described e.g. in [2], [3], [4].

The following text will give a brief presentation of a method for developing of algorithmic thinking of our students. This development should be enabled thanks to the method of the system approach applied in the subject *Algorithms*, which contributes to the motivation of students.

A. Principles of the system approach

Education at secondary schools and colleges in the area of informatics is directed mainly to a user's attitude. Only students who have attended optional subjects dealing with programming languages are familiar with creating algorithms. Thus a lot of students studying at our university lack any algorithmic knowledge at the beginning of their studies.

The system approach to the problem solution is such an approach which understands the studied phenomena and processes in complex internal and external contexts [6]. The system approach in education consists of the formulation, understanding and solving the problem, relevant processes, phenomena and processes that exist objectively in the outside world, and their transformation into a model educational situation.

The methodology, proposed in this paper, uses for training of algorithms the exercises which are based on modelling of real and idealized phenomena, processes and procedures. The methodology is intended to lead the student to a problem definition, to a creation of suitable models and to implementation and realization of the model through the principles of algorithms, i.e. the student should be able to determine about the input and output data and the procedure of the transformation of the data input into the data output under the system approach.

B. *Application of the system approach*

The proposed new methodological approach to teaching of algorithms can be applied:

1) In the field of sciences - in case of models of physical and chemical processes and phenomena;
2) In the technical field - in case of the process control of machines and simple robots;
3) In the humanities and social studies - in case of the processes and phenomena associated with this issue;
4) In management - in case of the management processes and quality control processes.

Mathematical calculus is essential in a “model” method, but only as a secondary issue. The standard method of a simple rewriting of mathematical expressions and equations into algorithms and then into the programs is maximally eliminated while using this approach.

It should be emphasized that the use of the system learning of algorithms based on a comprehensive methodological approach can be connected with the fact that mathematics, sciences and technology cannot do without the use of the ICT technology; new technologies have become an integral part. A current teacher has to master not only teaching his/her discipline, but also the foundations of the computer science. Likewise, a teacher of informatics (a computer programmer) cannot be only a computer teacher (a programmer), but s/he has to be able to systematically apply her/his knowledge of ICT in other fields.

The current non-systematic approach to teaching leads to such situations in which subjects are taught in isolation, without inter-disciplinary linkages.

The case study of application the system approach in the field of the humanities and social studies will be presented in the following part of the paper. It illustrates a step by step using of algorithm development and programming in the cryptoanalysis of monoalphabetical substituted cipher text as a suitable example of an algorithmic thinking development and also as an example of the interdisciplinary system learning that is useful to be included in secondary school teachers´ education.

III. Monoalphabetical substitution

The monoalphabetical substitution cipher is a cipher where one-to-one mapping is used to substitute each of the characters of the plaintext by a corresponding character of the ciphertext. A plaintext is a message before encryption. A plaintext is a normal text, e.g. in English, presented in a readable and understandable form. A ciphertext is an unreadable message after encryption. A ciphertext looks like a random jumble of letters or other characters. One way how to obtain a ciphertext from a plaintext is a character by character substitution realised according to the so-called conversion matrix. The conversion from a ciphertext back into a plaintext is called decryption. The decryption is realized by using the same conversion table in the opposite direction.

A. *Principles of monoalphabetical substitution*

A concrete example of a historic cipher is the substitution used in the Old Testament, where re-writers of the Bible wanted to leave their mark in the text. Sesah appeared instead of Babel. This encryption emerged from the substitution of the letters from the beginning of the Hebrew alphabet by letters from its end, namely the **aleph** by **tav**, **bet** by **shin** and **kaf** by **lamed**. Since the Hebrew text is often recorded only in consonants and vowels are added from the context, this cipher is known as the **Atbash** [6], [7].

Applying the same principle to the Roman alphabet (the international alphabet with 26 letters without diacritics is used), the *conversion table* is presented in Table 1.

A	B	C	D	E	F	G	H	I	J	K	L	M
Z	Y	X	W	V	U	T	S	R	Q	P	O	N

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M	L	K	J	I	H	G	F	E	D	C	B	A

Table 1 Conversion table

The Atbash type system has another advantage - the same conversion table can be used for both decryption and encryption. An additional advantage is that the table can be reduced by a half of the original conversion table.

A Conversion table of monoalphabetic substitutions, however, may be more general. For example, mixing the letters of the cipher alphabet according to the password can be used. It is shown in Table 2 the, where the password is based on the name of our university, from which repeated letter have been removed:

UNIVERSTY OF HADC KL (University of Hradec Kralove)

A	B	C	D	E	F	G	H	I	J	K	L	M
U	N	I	V	E	R	S	T	Y	O	F	H	A

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	C	K	L	B	G	J	M	P	Q	W	X	Z

Table 2 Conversion table with the password mixing letters

The question is how many monoalphabetic ciphers can be created. Each cipher can be described by the table, which has in the first row the characters of the alphabet sorted in the usual alphabetical order (A, B, C, ..., Z) and in the second row there are the letters arranged in any random order. Number of different orders, i.e. permutations of the 26 letters of the alphabet is

26! = 26.25.24. ... 3.2. 1 =
= 403291461126605635584000000,
approximately 403 quadrillions.

The fact that each single substitution is relatively easy to solve, was probably discovered for the first time by Arab scholar Al-Kindí in the 9th century AD.

Al-Kindi described, in two brief paragraphs of the manuscript, the principle of the frequency analysis, which is one of the most important tools of the classical cryptoanalysis. The origin of the frequency analysis is likely to have been influenced by studying of the Qur'an, which was so thorough that it not only examined the frequency of individual words, but even the frequency of individual letters. In Arabic, the letters ***a*** and ***l*** appear very often, while the frequency of ***j*** is about ten times lower. Frequency, or percentage distribution of individual letters, however, is different in each language.

The Table 3 shows the percentage distribution for English [8].

A	B	C	D	E	F	G	H	I	J	K	L	M
8,2	1,5	2,8	4,3	**12,7**	2,2	2,0	6,1	**7,0**	0,2	0,8	4,0	2,4
N	**O**	P	Q	R	S	**T**	U	V	W	X	Y	Z
6,7	**7,5**	1,9	0,1	6,0	6,3	**9,1**	2,8	1,0	2,4	0,2	2,0	0,1

Table 3 Percentage distribution of letters in a purely English text

The letters E, T, A, O and I are the most frequent letters in English. The longer a ciphertext is and the purer English is used in the text (without a mixture of foreign words), the more helpful the frequency analysis is. The procedure of the decryption will be described in the following text of the paper.

B. Principles of deciphering of the monoalphabetic substitution

First, the frequency analysis of the ciphertext has to be performed. If the text is electronically available, it is possible to use the tool on the Web [4]. Letters with the highest frequency are probably E, T, A, O and I - vowels. If the ciphertext is shorter, it is probable that the sequence of letters above will be replaced. Therefore other linguistic facts, knowledge and skills have to be included in the decryption analysis:

- Consonants and vowels are in the plaintext fairly regularly alternated;
- Doubling of the letters is typical of certain sounds;
- The frequency of pairs of consecutive letters, which are called bigrams;
- The ability to estimate certain words of the ciphertext, especially when it is possible because of being aware of the topic of the ciphertext.

Note: If all characters of the ciphertext have a similar frequency without significant maximums and minimums, it would probably be a polyalphabetic cipher. In these cases the solution has to be provided by other methods, which exceed the scope of this paper.

C. An example of hand deciphering of monoalphabetic substitution

The process of deciphering of the ciphertext encrypted by the monoalphabetical substitution will be shown in the following example:

An English ciphertext:

```
UOMEO MUWKP UMEPX FEMPJ WLXWE
AOYKP YRMJU MEPPF EMUXF JBUOM
JMPJW LXPFF BWNFI WPFPP FKMXS
WERKW LFSSM MYDGF JPMJK FEUWJ
JMEKP JMMPY EUWKO YERPF EDWJB
MPDFK PFSPO MFPOM JIQKY EMKKM
KFSPO WPWJM WUMJM GJFAQ LMOFQ
KMKKF POWPI MKYAM KOWTY ERSJM
MLFSS MMPFA JYEBY EPOMP WKPYE
RAMGW JPDME PPJWL XOWAS JMMSJ
QYPWE ATMRM PWICM KPFMW P
```

A performed frequency analysis gives the following results:

- the message has 271 characters;
- the most frequent letter is **M**, its occurrence is 41, the relative frequency of **M** is 15.1 %;
- the characters in the following order are **P** – 12.5 %, **W** - 8.5% and **F** - 8.1%.

Based on the above mentioned, the characters can be replaced by using the percentage distribution (Table 3) by the characters **E**, **T**, **A** and **O**. The remained characters will be replaced by the dash yet.

Moreover, the following information is to be given:

- the group of six letters **LFSSMM** occurs twice in the ciphertext;
- other information from another source is that the word **COFFEE** appears in the message;

Based on this information, we will try to replace the letters **L** and **S** by letters **C** and **F**.

- another group repeated twice is **SJMM**;
- after the above mentioned replacement it can be read **F**? **EE**.

It seems that letter **J** can be replaced by letter **R**.

We can see that the alternation of vowels and consonants is correct, so we can try to identify other vowels´ characters.

They will be marked by an asterisk and a cross:

```
--E-- E-A-T -E-T+ O-ETR AC+A-
--*-T *-ER- E-TTO -E-+O R---E
RETRA C+TOO -A-O- ATOTT O-E+F
A---A COFFE E*--O RTER- O--AR
RE--T REET* --A-- *--TO --AR-
ET-O- TOFT- EOT-E R---* -E—E-
-OFT- ATARE A-ERE -RO-- CE-O-
-E--O T-AT- E-*-E --A-* --FRE
ECOFF EETO- R*--* -T-ET A-T*-
--E-A RT-E- TTRAC +-A-F REEFR
-*TA- --E-E TA--E -TOEA T
```

We will try to replace the asterisk and the cross by vowels, a reasonable substitution seems to be **Y** by **I** and **X** by **Y**:

```
--E-- E-A-T -E-TY O-ETR ACYA-
--I-T I-ER- E-TTO -E-YO R---E
RETRA CYTOO -A-O- ATOTT O-EYF
A---A COFFE EI--O RTER- O--AR
RE--T REETI --A-- I--TO --AR-
ET-O- TOFT- EOT-E R---I -E--E
-OFT- ATARE A-ERE -RO-- CE-O-
-E--O T-AT- E-I-E --A-I --FRE
ECOFF EETO- RI--I -T-ET A-TI-
--E-A RT-E- TTRAC Y-A-F REEFR
-ITA- --E-E TA--E -TOEA T
```

An incomplete conversion table is shown in Table 4.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
W		L		M	S			Y	W		L			F					P					X	

Table 4 Incomplete conversion table

The name Tracy appears repeatedly in the plaintext, which is the name of the main character Tracy's Tiger book by William Saroyan. Let's try to use the writer's name as a password in such a way that the letters used repeatedly are deleted. We get *WILAMSROYN*. This ordering is in correspondence with the complete conversion table – see table 5:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
W	I	L	A	M	S	R	O	Y	N	B	C	D	E	F	G	H	J	K	P	Q	T	U	V	X	Z

Table 5 Complete conversion table

The encrypted plaintext can be got from the ciphertext by using of the above given table:

```
WHENH EWAST WENTY ONETR ACYAN
DHIST IGERW ENTTO NEWYO RKWHE
RETRA CYTOO KAJOB ATOTT OSEYF
ANGSA COFFE EIMPO RTERS ONWAR
RENST REETI NWASH INGTO NMARK
ETMOS TOFTH EOTHE RBUSI NESSE
SOFTH ATARE AWERE PRODU CEHOU
SESSO THATB ESIDE SHAVI NGFRE
ECOFF EETOD RINKI NTHET ASTIN
GDEPA RTMEN TTRAC YHADF REEFR
UITAN DVEGE TABLE STOEA T
```

The final step is to separate the pentads and to insert the spaces to make the text readable. The final text is as follows:

"*When he was twenty-one Tracy and his tiger went to New York, where Tracy took a job at Otto Seyfang's, a coffee importer's on Warren Street in Washington Market. Most of the other businesses of that area were produce houses, so that besides having free coffee to drink - in the Tasting Department - Tracy had free fruit and vegetables to eat.*"

It is a *passage* from the book Tracy's Tiger by William Saroyan. We can see from the example that the classic hand encryption of the monoalphabetic substitution is not an easy thing. The combination of the mathematical methods (frequency analysis) with the linguistic method (the estimation of the plaintext on the basis of the estimated words) has been used.

IV. Automation of the Cryptoanalysis

The following part of the paper presents a description of a possibility of using mathematical methods and algorithms for a cryptanalysis of the ciphertext.

The algorithm described in the following text, is a basis of the computer program, which automatically decrypts any ciphertext encrypted by the monoalphabetical substitution cipher. The original form of this algorithm is taken from the paper by Thomas Jacobsen [9]. Unlike this form, our algorithm was applied to a Czech text with no spaces.

The first task is to obtain a suitable reference text of the bigrams rate in the Czech language. The text from the book The Gardener's Year by Karel Capek, was used. It has about 99,000 letters. In the text we ignored the diacritics, punctuations and spaces and for each pair of consecutive letters, we incremented the square matrix element of dimension 26 (a number of international alphabet letters). Finally, the whole matrix is multiplied by 100 and divided by the number of letters of the reference text, so the value of the matrix elements now represents the percentual relative frequency of bigrams.

The developed program enables easy changing of the values of the reference matrix, so it can be used for other languages than Czech.

A. Algorithm

The algorithm itself was divided into three relatively independent procedures.

Frequency:

5) This procedure performs an initial setup for the conversion table. The frequency of each character in the ciphertext is detected and ciphertext characters are sorted in the descending order of frequency. That determined the sum of the rows from the reference matrix ***E*** that corresponds to the frequencies of letters in the reference text; the reference text characters are sorted in the descending order of frequency. Finally, the procedure pairs the first most frequent letter of the ciphertext with the first most frequent letter of the reference text, then the second most frequent letter of the ciphertext with the second most frequent letter of the reference text, and so on.
6) The procedure creates matrix ***D*** of relative frequencies of the bigrams of the ciphertext and evaluates the compliance with the reference text by using an evaluation function:

$$f = \sum_{i=1}^{26} \sum_{j=1}^{26} \left| D_{ij} - E_{ij} \right| \tag{1}$$

Bigrams:

1) Step by step, 2 columns and 2 rows in the matrix ***D*** are exchanged and thus the matrix ***D'*** is obtained. The columns and rows are exchanged in the order of the frequencies of the letters in the ciphertext from the most frequent to the least frequent:
   - the column corresponding to the order of the first exchanged character is replaced by the column corresponding to the second exchanged character;
   - then the row corresponding to the order of the first exchanged character is replaced by the row corresponding to the second exchanged character;
2) The substitution of the columns and rows of the matrix corresponds to the replacement of the letters in the second row of the conversion table. First, the first and the second letters are replaced then the first and the third letters are replaced and so on, until finally the first letter is exchanged by the last one. The same is done with the second letter, which is replaced with the third one, then with the forth one and so on. Finally next to last letter is replaced by last one.
3) After each substitution a new matrix ***D'*** is obtained, and the evaluation of the compliance of the relative frequency of the bigrams in the ciphertext and the reference text is obtained using the evaluation function f':

$$f' = \sum_{i=1}^{26} \sum_{j=1}^{26} \left| D'_{ij} - E_{ij} \right| \tag{2}$$

4) After each substitution the values f and f' are compared by inequality:
   $$f' > f \tag{3}$$
5) If $f' > f$, the procedure continues with the next pair of the letters in order. However, if $f' < f$, the procedure immediately stops the process of the letter substitution and the exchange in the conversion table is proposed, which will improve the compliance (lowering the value of evaluation function f). If the program overruns all the possible exchanges of the characters, without any condition $f' < f$, the message is notified to the operator. If the ciphertext is not fully deciphered, the operator can propose a manual replacement.

Exchange:

1) As not always the procedure ”Bigrams“ suggests an optimal exchange, especially for short texts (less than 500 characters), it is possible to perform the required exchange manually, to set it directly into the conversion table based on the reading of a partly deciphered text and estimating of the content of messages and expected words.
2) Whether the exchange is proposed automatically or manually, the procedure ”Exchange“ will create a new matrix ***D*** of relative frequencies of the bigrams of the ciphertext, and it will provide a new assessment of compliance with the reference text using the evaluation function $f\,(2)$.

From the description of the above procedures it is obvious that the procedure ”Frequency“ is run only once at the beginning of the program to set the appropriate initial conditions. The procedures ”Bigrams“ and ”Exchange“ are run alternately, or instead of run the procedure ”Bigrams“, a manual exchange of characters in the second row of the conversion table is suggested based on the reading of the text and partly decrypted ciphertext and based on the estimation of the words contained in the plaintext that we get.

V. Computer Simulation of the Algorithm in MS Excel

A. Programming in Visual Basic for MS Excel

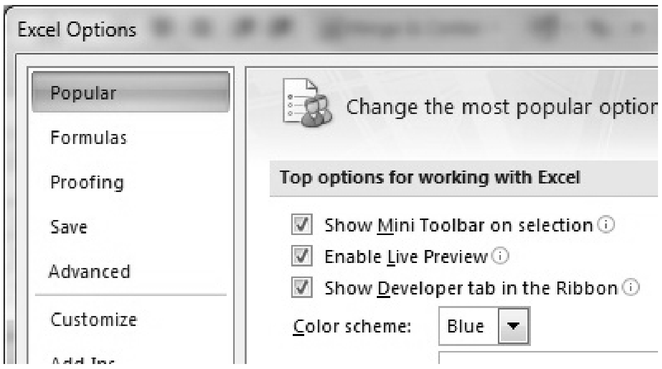
All MS Office application like MS Excel, MS Access, MS Word, MS PowerPoint, MS Outlook and others, have implemented a programming language *Visual Basic for Application* (*VBA*). It is an object-oriented programming language, which is based on Visual Basic.

VBA contains the same commands as Visual Basic and extra commands that operate with objects of their own parent application (with cells, selection, sheets, etc. in MS Excel). This can be regarded as the main advantage, especially in teaching of programming. Every user of MS Office has a programming language, and s/he can try to use it as s/he likes, s/he can create his/her own ”little useful application”, which can become the basis on which a further interest in programming develops.

VBA, although it is an object-oriented language, allows to handle the basic algorithmic tasks based on fully structured, sequential approach, which is especially helpful in the initial stages of algorithm development and programming.

B. *VBA Integrated development environment*

The integrated development environment (IDE) of VBA is available in MS Excel 2007 in the ribbon tab ”Developer“. This tab is not displayed by default. The tab “Developer” in version of MS Excel 2007 can be activated in the ”*Excel Options*“ dialog box (which is reached by clicking on the *Excel Option* button in the main menu) in the folder ”*Popular*“ – see Figure 1. The ”*Show Developer tab in the Ribbon*“ checkbox has to be checked.


Excel Options

Popular
Formulas
Proofing
Save
Advanced
Customize
Add-Ins

Change the most popular optior

Top options for working with Excel

☑ Show Mini Toolbar on selection
☑ Enable Live Preview
☑ Show Developer tab in the Ribbon
Color scheme: Blue

Fig. 1 Activation of Developer tab

To be able to exploit this IDE, the macro security has to be set to low. In MS Excel 2007, this setting can be performed on the “Developer tab”, “Macro Security” command, where the option button “Enable all macros” has to be checked.

The IDE can be opened in two ways - either through the “Developer tab”, click on “Visual Basic” command, or by pressing the keys Alt-F11.

The IDE has the Project Explorer in its left part, which shows a list of currently open projects - Workbooks, Worksheets, Modules and Forms.

In the case of sequential structured programming, the programs are saved in and run from separate *Modules*. One module can contain more programs.

In the case of object-oriented programming, the *Graphical user interface elements* (*GUI*) are inserted in the *Form*. VBA includes a reduced number of GUI like Command_Button, Text_Box, Label, Option_Button and Check_Box.

C. *Inputs / Outputs in VBA for MS Excel*

The VBA for MS Excel gives different possibilities of Inputs and Outputs of variables.

Inputs:

1) The command A = InputBox(„Input A“) opens the input dialog box into which the user inputs the value of the variable A.
2) Command A = Cells (R, C) is the value of variable A read from the cell in row R and column C.
3) The TextBox can be used for reading the data in the object oriented approach of programming.

Outputs:

1) The command MsgBox(A) opens the output dialog box in which the value of variable A is output.
2) Command Cells (R, C) = A saves the value of variable A in a cell in row R and column C. This value can be used in following program executions.
3) The TextBox can be used for writing the data in the object oriented approach.

D. *Variables of array type*

Unlike the conventional programming languages like C #, Delphi, Visual Basic, etc., the VBA for MS Excel enables two ways of working with data stored in the form of vectors (one-dimensional variables) or matrixes (two-dimensional variables).

The first option is the same as for other languages, where the vector and matrix variables are declared in a section of variable declarations:

```
Dim A(1 To 26) As String            ‘Vector
Dim B(1 To 26, 1 To 26) As Long ‘Matrix
```

The second option is to store one- and two-dimensional data directly into cells in Excel worksheets. The command

```
Cells(2,3) = 10
```

can be understood as an element of the matrix with the row coordinates 2 and column coordinates 3 with the value 10. The advantage of this approach with the vector and matrix data is that the values of individual elements of the vector / matrix are visible to the user.

VI. MONOALPHABETICAL SUBSTITUTION SOLVED IN MS EXCEL

The solution of the above mentioned algorithmic task in MS Excel is subdivided into a number of more or less independent parts.

A. *Analysis of the reference text*

As already mentioned, the basic requirement set to the task is the applicability of deciphering a monoalphabetical cipher text in any selected language. For this reason, our application analyzes a ”rather long“ and “poor language” text containing no foreign words. The analysis of this reference text gives the first information about the frequency of individual letters, as well as the information of the frequency of bigrams of the selected language.

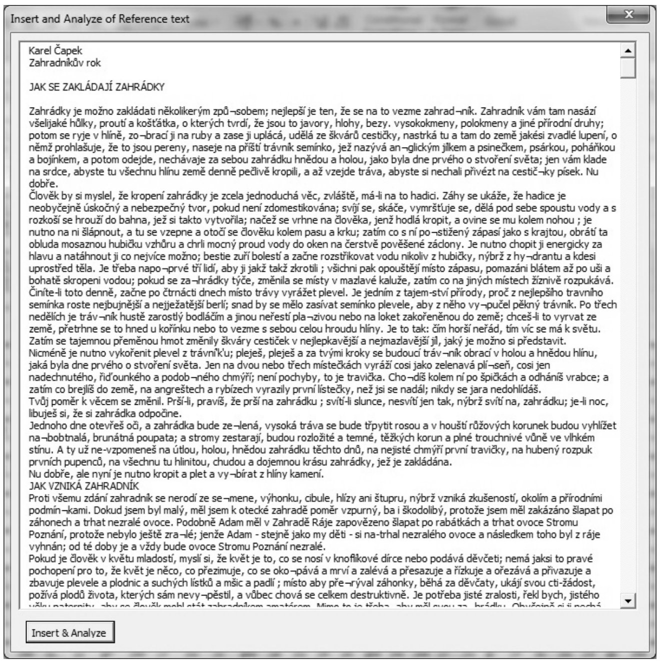
The procedure of the analysis of the reference text can be split into the following parts:

1) Downloading of the reference text;
2) Correction of the reference text;
3) Frequency analysis of individual letters;
4) Ordering of letters according to descendent frequency;
5) Creation of the reference bigram matrix *E*;
6) Output of letter frequency and bigram matrix;

1. Downloading of a reference text

The Form with two objects - Textbox and a Command button - is assembled for reading of the reference text. The form is shown in Figure 2.

One of the TextBox's properties is “Multiline”. In this case, reading a long text, this property is set to “True” (default is “False”).


Insert and Analyze of Reference text

Karel Čapek
Zahradníkův rok

JAK SE ZAKLÁDAJÍ ZAHRÁDKY

Zahrádky je možno zakládati několikerým způ¬sobem; nejlepší je ten, že se na to vezme zahrad¬ník. Zahradník vám tam nasází všelijaké hůlky, proutí a košťátka, o kterých tvrdí, že jsou to javory, hlohy, bezy. vysokokmeny, polokmeny a jiné přírodní druhy; potom se ryje v hlíně, zo¬brací ji na ruby a zase ji uplácá, udělá ze škvárů cestičky, nastrká tu a tam do země jakési zvadlé lupení, o němž prohlašuje, že to jsou pereny, naseje na příští trávník semínko, jež nazývá an¬glickým jílkem a psinečkem, psárkou, poháňkou a bojínkem, a potom odejde, nechávaje za sebou zahrádku hnědou a holou, jako byla dne prvého o stvoření světa; jen vám klade na srdce, abyste tu všechnu hlínu země denně pečlivě kropili, a až vzejde tráva, abyste si nechali přivézt na cestič¬ky písek. Nu dobře.
Člověk by si myslel, že kropení zahrádky je zcela jednoduchá věc, zvláště, má-li na to hadici. Záhy se ukáže, že hadice je neobyčejně úskočný a nebezpečný tvor, pokud není zdomestikována; svíjí se, skáče, vymršťuje se, dělá pod sebe spoustu vody a s rozkoší se hrouží do bahna, jež si takto vytvořila; načež se vrhne na člověka, jenž hodlá kropit, a ovine se mu kolem nohou ; je nutno na ni šlápnout, a tu se vzepne a otočí se člověku kolem pasu a krku; zatím co s ní po¬stižený zápasí jako s krajtou, obrátí ta obluda mosaznou hubičku vzhůru a chrlí mocný proud vody do oken na čerstvě pověšené záclony. Je nutno chopit ji energicky za hlavu a natáhnout ji co nejvíce možno; bestie zuří bolestí a začne rozstřikovat vodu nikoliv z hubičky, nýbrž z hy¬drantu a kdesi uprostřed těla. Je třeba napo¬prvé tří lidí, aby ji jakž takž zkrotili ; všichni pak opouštějí místo zápasu, pomazáni blátem až po uši a bohatě skropeni vodou; pokud se za¬hrádky týče, změnila se místy v mazlavé kaluže, zatím co na jiných místech žíznivě rozpukává.
Činíte-li toto denně, začne po čtrnácti dnech místo trávy vyrážet plevel. Je jedním z tajem-ství přírody, proč z nejlepšího travního semínka roste nejbujnější a nejježatější berlí; snad by se mělo zasívat semínko plevele, aby z něho vy¬pučel pěkný trávník. Po třech nedělích je tráv¬ník hustě zarostlý bodláčím a jinou neřestí pla¬zivou nebo na loket zakořeněnou do země; chceš-li to vyrvat ze země, přetrhne se to hned u kořínku nebo to vezme s sebou celou hroudu hlíny. Je to tak: čím horší neřád, tím víc se má k světu.
Zatím se tajemnou přeměnou hmot změnily škváry cestiček v nejlepkavější a nejmazlavější jíl, jaký je možno si představit.
Nicméně je nutno vykořenit plevel z trávní'k'u; pleješ, pleješ a za tvými kroky se budoucí tráv¬ník obrací v holou a hnědou hlínu, jaká byla dne prvého o stvoření světa. Jen na dvou nebo třech místečkách vyráží cosi jako zelenavá plí¬seň, cosi jen nadechnutého, řiďounkého a podob¬ného chmýří; není pochyby, to je travička. Cho¬díš kolem ní po špičkách a odháníš vrabce; a zatím co brejlíš do země, na angreštech a rybízech vyrazily první lístečky, než jsi se nadál; nikdy se jara nedohlídáš.
Tvůj poměr k věcem se změnil. Prší-li, pravíš, že prší na zahrádku ; svítí-li slunce, nesvítí jen tak, nýbrž svítí na, zahrádku; je-li noc, libuješ si, že si zahrádka odpočine.
Jednoho dne otevřeš oči, a zahrádka bude ze¬lená, vysoká tráva se bude třpytit rosou a v houští růžových korunek budou vyhlížet na¬bobtnalá, brunátná poupata; a stromy zestarají, budou rozložité a temné, těžkých korun a plné trouchnivé vůně ve vlhkém stínu. A ty už ne-vzpomeneš na útlou, holou, hnědou zahrádku těchto dnů, na nejisté chmýří první travičky, na hubený rozpuk prvních pupenců, na všechnu tu hlinitou, chudou a dojemnou krásu zahrádky, jež je zakládána.
Nu dobře, ale nyní je nutno kropit a plet a vy¬bírat z hlíny kamení.
JAK VZNIKÁ ZAHRADNÍK
Proti všemu zdání zahradník se nerodí ze se¬mene, výhonku, cibule, hlízy ani štupru, nýbrž vzniká zkušeností, okolím a přírodními podmín¬kami. Dokud jsem byl malý, měl jsem k otecké zahradě poměr vzpurný, ba i škodolibý, protože jsem měl zakázáno šlapat po záhonech a trhat nezralé ovoce. Podobně Adam měl v Zahradě Ráje zapovězeno šlapat po rabátkách a trhat ovoce Stromu Poznání, protože nebylo ještě zra¬lé; jenže Adam - stejně jako my děti - si na-trhal nezralého ovoce a následkem toho byl z ráje vyhnán; od té doby je a vždy bude ovoce Stromu Poznání nezralé.
Pokud je člověk v květu mladostí, myslí si, že květ je to, co se nosí v knoflíkové dírce nebo podává děvčeti; nemá jaksi to pravé pochopení pro to, že květ je něco, co přezimuje, co se oko¬pává a mrví a zalévá a přesazuje a řízkuje a ořezává a přivazuje a zbavuje plevele a plodnic a suchých lístků a mšic a padlí ; místo aby pře¬rýval záhonky, běhá za děvčaty, ukájí svou cti-žádost, požívá plodů života, kterých sám nevy¬pěstil, a vůbec chová se celkem destruktivně. Je potřeba jisté zralosti, řekl bych, jistého

Fig. 2 Form for reading and analyze of a reference text

Another property is “Text”. This property is modified when a user inputs the text in the Textbox. This property is a String data type, which enables to input the text with the length of approximately two billion characters.

2. Correction of the reference text

After pressing the button, the ”unwanted“ characters (spaces, commas, dots, other punctuation marks etc.) are first filtered out to get a clear text with the characters from “A” to “Z” (26 characters in total).

3. Frequency analysis of individual letters

The array variables (instead of the cells matrix variables) are used for the analysis of the reference text with regard to the speed of calculation. The frequency analysis of each letter is made so that the whole text is analyzed character by character in the following cycle:

```
For I = 1 To Len(RefText)
  OrdCh = Asc(Mid(RefText, I, 1))
  Ch(OrdCh) = Ch(OrdCh) + 1
Next I
```

The reference text is saved to variable RefText. Array Ch[1..26] of the integer type gradually increments. For example, for letter “A” is carried out the expression Ch[1] = Ch[1]+1, for letter “B” Ch[2] = Ch[2]+1, and so on.

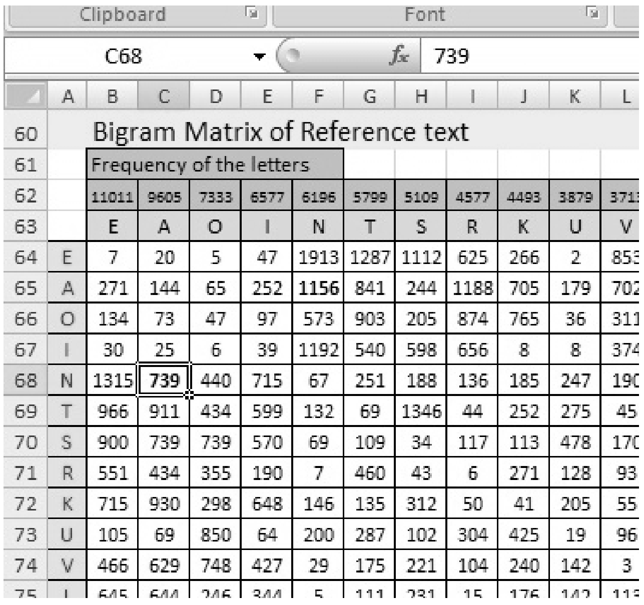
4. Ordering of letters according to descendent frequency

After the frequency analysis, the array Ch[1..26] is sorted in the descendent order with help of the “bubble sort” method:

```
For I = 26 To 2 Step -1
  For J = 26 To 27 - I + 1 Step -1
    If Ch(J) > Ch(J - 1) Then
      Tmp = Ch(J)
      Ch(J) = Ch(J - 1)
      Ch(J - 1) = Tmp
    End If
  Next J
Next I
```

5. Creation of the reference bigram matrix *E*

The matrix of bigrams *E* of the reference text will be created in the next step. The array E [1. .26, 1 .. 26] of the integer type is used. The columns of the matrix (the second index of E) determine the first letter of bigrams, the rows (the first index of E) determine the second letter of bigrams. The situation is shown in Figure 3.



	A	B	C	D	E	F	G	H	I	J	K	L
60		Bigram Matrix of Reference text										
61		Frequency of the letters										
62		11011	9605	7333	6577	6196	5799	5109	4577	4493	3879	371
63		E	A	O	I	N	T	S	R	K	U	V
64	E	7	20	5	47	1913	1287	1112	625	266	2	85
65	A	271	144	65	252	1156	841	244	1188	705	179	70
66	O	134	73	47	97	573	903	205	874	765	36	311
67	I	30	25	6	39	1192	540	598	656	8	8	374
68	N	1315	739	440	715	67	251	188	136	185	247	19
69	T	966	911	434	599	132	69	1346	44	252	275	45
70	S	900	739	739	570	69	109	34	117	113	478	17
71	R	551	434	355	190	7	460	43	6	271	128	93
72	K	715	930	298	648	146	135	312	50	41	205	55
73	U	105	69	850	64	200	287	102	304	425	19	96
74	V	466	629	748	427	29	175	221	104	240	142	3
75	[illegible]	645	644	246	344	5	111	231	15	176	142	[illegible]

Fig. 3 Bigram matrix of reference text

From the figure it is clear, that bigram AN appears 739 times in our text, while the bigram NA appears 1156 times. The characters in rows and columns are sorted out through a frequency analysis made in the previous step. The part of the VBA code, which creates the matrix *E,* is as follows:

```
For I = 1 To Len(RefText) - 1
  Ch1 = Mid(RefText, I, 1)
  Ch2 = Mid(RefText, I + 1, 1)
  R = 1
  C = 1
  Do Until Od(C) = Ch1
    C = C + 1
  Loop
  Do Until Od(R) = Ch2
    R = R + 1
  Loop
  E(R, C) = E(R, C) + 1
Next I
```

The variables R and C determine indexes of matrix, the variable Od(I) of char type determines character in the given column and row, e.g. Od(1)="E", Od(2)="A", ETA in our case.

6. Storage of the frequency of letters and bigram matrix

The last step is the storage of the matrix *E* to MS Excel „Bigrams" worksheet in the range A62:AA89. The first coordinates of the matrix E[I,J] correspond to rows and the second coordinates correspond to the columns in the Excel worksheet.

In the next parts of the program there will be used only the values of the reference matrix *E* saved in the MS Excel worksheet, titled „Bigrams"; variables Ch[I] and E(I,J) will be released from the memory of the computer..

B. Downloading and analyzing of the cipher text

The next step is downloading and analyzing of the cipher text.

This step is implemented in VBA, like in the previous step, by simple Form with Textbox and Command button. After pressing the Command button, the following parts of the procedure are executed:

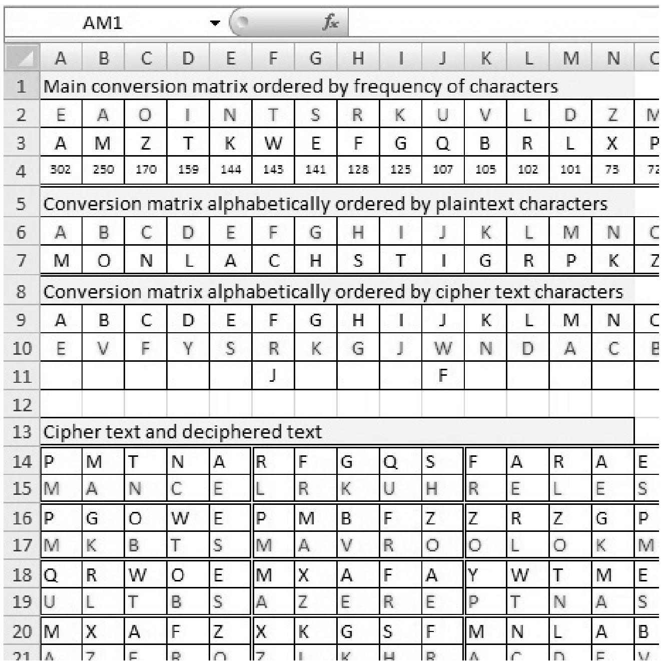
1) Downloading of the cipher text;
2) Correction of the cipher text;
3) Output of the cipher text to cells of MS Excel worksheet called "Main".

This procedure is realized by following programming code:

```
…
CiphText = TextBox1.Text
Row = 13
Col = 1
For I = 1 To Len(CiphText)
  If (Col - 1) / 50 = Int((Col - 1) / 50) Then
    Row = Row + 1
    Col = 1
  Else
    Col = Col + 1
  End If
Cells(Row,Col) = Mid(CiphText,I,1)
Next I
…
```

This procedure also includes commands for formatting of the cell borders and separating of the cells by petitions of the letters – see Figure 4.

In the cells of the even rows there is a cipher text, in the corresponding cells of the odd rows the decrypted text will be displayed in the next steps of the program.



AM1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Main conversion matrix ordered by frequency of characters													
2	E	A	O	I	N	T	S	R	K	U	V	L	D	Z
3	A	M	Z	T	K	W	E	F	G	Q	B	R	L	X
4	302	250	170	159	144	143	141	128	125	107	105	102	101	73
5	Conversion matrix alphabetically ordered by plaintext characters													
6	A	B	C	D	E	F	G	H	I	J	K	L	M	N
7	M	O	N	L	A	C	H	S	T	I	G	R	P	K
8	Conversion matrix alphabetically ordered by cipher text characters													
9	A	B	C	D	E	F	G	H	I	J	K	L	M	N
10	E	V	F	Y	S	R	K	G	J	W	N	D	A	C
11						J				F				
12														
13	Cipher text and deciphered text													
14	P	M	T	N	A	R	F	G	Q	S	F	A	R	A
15	M	A	N	C	E	L	R	K	U	H	R	E	L	E
16	P	G	O	W	E	P	M	B	F	Z	Z	R	Z	G
17	M	K	B	T	S	M	A	V	R	O	O	L	O	K
18	Q	R	W	O	E	M	X	A	F	A	Y	W	T	M
19	U	L	T	B	S	A	Z	E	R	E	P	T	N	A
20	M	X	A	F	Z	X	K	G	S	F	M	N	L	A

Fig. 4 The "Main" sheet with conversion matrixes and petitions of the cipher and the deciphered text

C. Frequency analysis of the cipher text

The frequency analysis of the cipher text is carried out by the separate procedure Frequency, written in the VBA module. It is a structured sequential program. This procedure, as described above, is executed once and determines the frequency of each letter in the cipher text. This procedure can be subdivided into several steps:

1) Initialization of the conversion matrix;
2) Frequency analysis;
3) Ordering of letters according to descendent frequency;
4) Output and storage of the conversion matrixes;
5) Call of the procedure Create_Bigrams

1. Initialization of the conversion matrix

On the Excel "Main" worksheet three conversion matrixes are displayed for clarity. In rows 2 and 3, the matrix is sorted by descending order of letters frequency. In rows 2, the letters are arranged according to the frequency analysis of the reference text.

The presented frequency analysis of the cipher text is carried out in rows 3 and 4. During initialization, row 3 is filled with the letters in alphabetical order ("A", "B", etc.) and row 4 is filled with zero.

2. Frequency analysis

The frequency analysis is provided by the following command rows of the procedure:

```
TmpCh = Asc(Cells(Row, Col)) – 64
Cells(4, TmpCh) = Cells(4, TmpCh) + 1,
```

where variables Row and Col assign actual row and column of the cell from which the character of the cipher is read. The function ASC() transfers this character to ASCII value.

When it is given that the ASC ("A") = 64, ASC (B) = 65 etc., then variable TmpCh = ASC () - 64 indicates the column in row 4, where the value determining the frequency of the given character is incremented.

3. Ordering of letters according to descendent frequency

In this step, the procedure arranges the letters of the conversion matrix according to descendent frequency. The bubble sort method is used to provide the ordering.

4. Output and storage the conversion matrixes

Subsequently, the procedure creates other two amended conversion matrixes. The first is arranged alphabetically by the plaintext characters – rows 6 and 7, the second is arranged alphabetically by the ciphertext characters – rows 9 and 10.

5. Call of the procedure New_Bigrams_Matrix_D

The final step is calling the procedure New_Bigrams_Matrix_D, which creates a matrix of relative frequencies of bigrams *D* – see the text below.

D. Initialization of matrix D

Since the matrix of bigrams *D* is created at the end of the procedure of Frequency, and thereafter also at the end of the procedure of Exchange, which is running if the condition (3) is fulfilled, the matrix *D* is generated by the same procedure of New_Bigrams_Matrix_D.

The matrix *D* is created and stored in the cells of the "Bigrams" sheet in the range A32:AA58. While the matrix *E* is created through using of the array variables, elements of the matrix *D* are calculated directly in cells on the "Bigrams" sheet. This approach is suitable for clearity reasons.

The procedure is divided into the following parts:

1) Creation of the main column and main row of the matrix;
2) Frequency analysis of the bigrams;
3) Calculation of the function f;

1. Creation of the main column and main row of the matrix

The main column, resp. main row of the matrix are copies of the first and the second rows of the main conversion matrix stored in the "Main" sheet The column contains the letters of the reference text arranged in the descending order according to the frequency; the row contains the letters of ciphertext arranged in the descending order according to the frequency of the ciphertext (procedure of Frequency).

2. Frequency analysis of the bigrams

The calculation of the values of the matrix elements D_{ij} is performed using the special sub-procedures due to the fact that mostly the same part of the program will be used also in the procedure of Bigrams (see below). Values of D_{ij} are calculated directly in cells on an Excel worksheet:

```
Public Sub Bigrams_Calculate(IniRow, IniCol, FCol)
For I = 1 To Len(CiphText) - 1
  Ch1 = Mid(CiphText, I, 1)
  Ch1 = Mid(CiphText, I + 1, 1)
  R = IniRow
  C = IniCol
  Do Until Cells(32, C) = Ch1
    C = C + 1
  Loop
  Do Until Cells(R, 1) = Ch2
    R = R + 1
  Loop
  Cells(R, C) = Cells(R, C) + 1
Next I
…
```

The input parameters of the sub-procedures of IniRow and IniCol determine the upper left corner of the matrix. The cipher text is stored in the variable CiphText of the string type, variables R and C determine the row and column of the matrix *D*. The incrimination of the bigrams cell appears in the last but one row.

3. Calculation of the function *f*

The matrix elements E_{ij} and D_{ij} are used for calculation of variable *f* given by (1). The sub-procedure of Bigrams_Calculate continues with the following code:

```
…
F = 0
For I = 1 To 26
  For J = 1 To 26
    Dij = Cells(I + 32, J + 1)
    Eij = Cells(I + 63, J + 1)
    F = F + Abs(Dij - Eij)
  Next J
Next I
Cells(4, FCol) = F
```

The value of *f* function is stored for further using in cell AC4, so FCol = 29.

The whole procedure of New_Bigrams_Matrix_D has the following code:

```
Sub New_Bigrams_Matrix_D ()
  For I = 1 To 26
    'Creation of main matrix column
    Cells(32 + I, 1) = Cells(1 + I, 1)
    'Creation of main matrix row
    Cells(32, I + 1) = Cells(1, I + 1)
  Next I
  'zero adjustment of Dij elements
  Range(Cells(33, 2), Cells(58, 27)) = 0
  Call Bigrams_Calculate (33, 2, 29)
  Cells(2, 29) =1 'initialization for bigrams exchange
  Cells(2, 30) =2 'initialization for bigrams exchange
End Sub
```

E. Analysis of the bigrams

The next procedure - Bigrams - analyses a text based on the frequency of the bigrams and creates a new conversion matrix. This procedure creates a matrix *D'* by a simultaneous substitution of two rows and two columns of the matrix *D*, calculates function *f'* (2) and evaluates the condition (3).

The matrix *D* ' is created and stored in the cells of the MS Excel on the “Bigrams” sheet in the range A1:AA27.

The procedure of Bigrams can be expressed by the flowchart shown in the Figure 5.

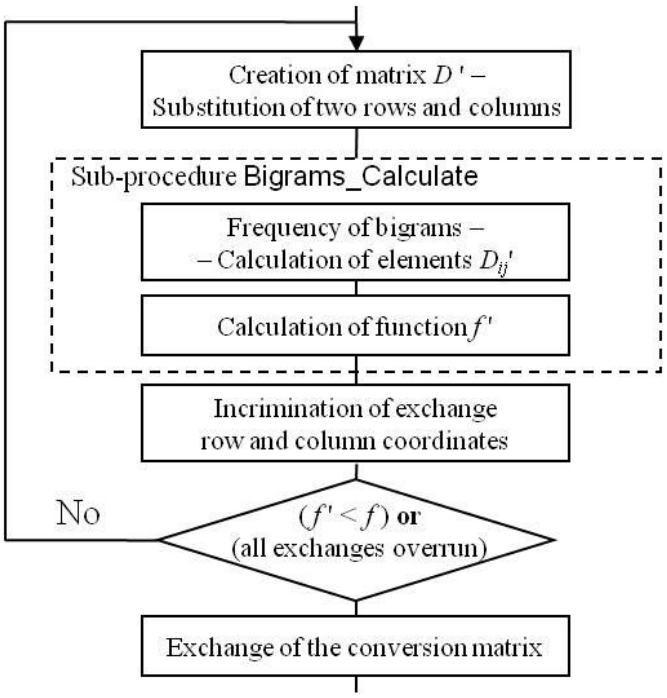

Creation of matrix D ' –
Substitution of two rows and columns
Sub-procedure Bigrams_Calculate
Frequency of bigrams –
– Calculation of elements D_{ij}'
Calculation of function f'
Incrimination of exchange
row and column coordinates
No
($f' < f$) or
(all exchanges overrun)
Exchange of the conversion matrix

Fig. 5 Procedure Bigrams expressed by flowchart

The blocks of the procedure *Frequency of bigrams* and *Calculation of function f* ', which are repeated in the loop with the condition, are identical to the previous procedure of New_Bigrams_Matrix_D.

The simplified procedure code is as follows:

```
Sub Bigrams()
Do
Ch1 = Cells(2, 29) ‘1. coordinates of the substitution
Ch2 = Cells(2, 30) ‘2. coordinates of the substitution
‘Exchange of the columns
   Tmp = Cells(1, Ch1 + 1)
   Cells(1, Ch1 + 1) = Cells(1, Ch2 + 1)
   Cells(1, Ch2 + 1) = Tmp
‘Exchange of the rows
   Tmp = Cells(Ch1 + 1, 1)
   Cells(Ch1 + 1, 1) = Cells(Ch2 + 1, 1)
   Cells(Ch2 + 1, 1) = Tmp
   Call Bigrams_Calculate (2, 2, 30)
‘Incrimination of the exchange coordinates
If Ch2 < 26 Then
     Cells(2, 30) = Cells(2, 30) + 1
   Else
     Cells(2, 29) = Cells(2, 29) + 1
     Cells(2, 30) = Cells(2, 29) + 1
  End If
Loop Until  (Cells(4, 30) < Cells(4, 29)) or _
         _ (Cells(2, 29) ≥ 26)
‘Exchange of the conversion matrix
For I = 1 To 26
   Sheets("Main").Cells(3, I) = Cells(1, I + 1)
Next I
End Sub
```

F. Exchange

The last procedure of Exchange proceeds the decryption of the cipher text through the conversion matrix directly on the ”Main“ sheet in the rows below the cipher text. The situation is shown in Figure 4, where the cipher text is in rows 14, 16, 18 etc. and the deciphered text is in rows 15, 17, 19 etc.

If the loop “until” in the procedure Bigrams is stopped by condition (Cells(2, 29) ≥ 26), all possible exchanges overrun. In this case the manual exchange has to be suggested. The manual exchange of “F” and “J” is suggested in the row 11 – see Figure 4.

Simultaneously, the procedure New_Bigram_Matrix_D is executed. Under this procedure it is possible to perform the required exchange manually in the row 11 of the “Main” sheet.

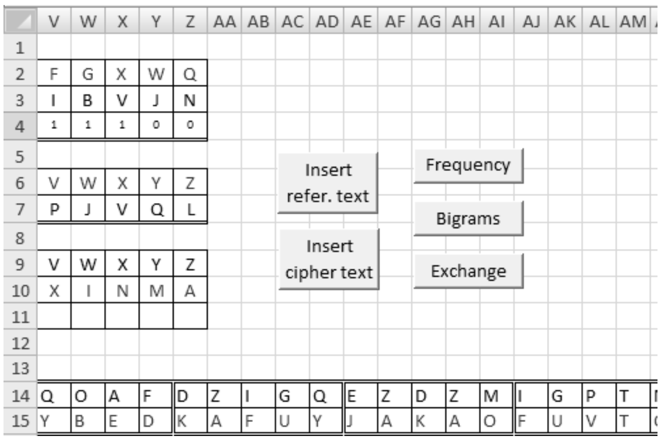
G. Evaluation of the MS Excel solution

That algorithm is very reliable and for sufficiently long texts (with a length of at least 500 characters) it usually operates completely automatically without a need of any manual correction.

The presented solution in MS Excel runs under the control of the operator through pressing of the command buttons in the following order:

Insert Reference text, *Insert Cipher text*, *Frequency* and repeatedly *Bigrams*, *Exchange*.

The command buttons are placed directly in the “Main” sheet– see Figure 6.



	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
1																		
2	F	G	X	W	Q													
3	I	B	V	J	N													
4	1	1	1	0	0													
5																		
6	V	W	X	Y	Z													
7	P	J	V	Q	L													
8																		
9	V	W	X	Y	Z													
10	X	I	N	M	A													
11																		
12																		
13																		
14	Q	O	A	F	D	Z	I	G	Q	E	Z	D	Z	M	I	G	P	T
15	Y	B	E	D	K	A	F	U	Y	J	A	K	A	O	F	U	V	T

Fig. 6 Command buttons for the user control of the program running

The possibility of the manual intervention in the process of solving is suitable for the decryption of short messages.

The application in MS Excel verified the functionality of the presented algorithm. As a reference text, the Czech text “The Gardeners’ Year” by Karel Capek was used. The length of this text is 98,817 characters. As a cipher text, the Czech text “Blue Chrysanthemum” by Karel Capek with the length of 2,483 characters was used.

This ciphered text is automatically deciphered in 7955 exchanges. The manual exchange is needed only once between “J” and “S”. Figure 7 shows the dependency of the function *f* on the *Number of Exchanges*.

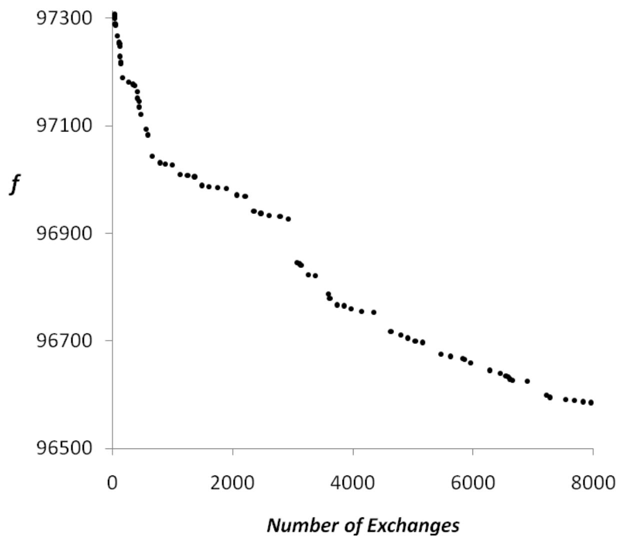


Fig. 7 Dependency of the function *f* on the *Number of Exchanges*

VII. Conclusion

There are various approaches how to provide training in algorithms, how to introduce and develop basic algorithmic thinking of students.

The paper offered one of the kinds of the possible teaching / learning strategies using the system approach. The system approach can be set as the default paradigm for a wide integration of the principles of the algorithm development into education. The paper emphasizes the fact that the algorithm is neither a mere mathematics, nor a mere programming.

Cryptoanalysis, concretely cryptoanalysis of monoalphabetical substitution, has been chosen as one possible example of how to present the application of the system approach.

Acknowledgment

This research has been partially supported by the Specific research project of the Faculty of Science of University of Hradec Kralove No. 2129.

References

[1] E. Milkova, “Multimedia Applications and their Benefit for Teaching and Learning at Universities”, *WSEAS Trans. on Information Science & Applications,* Issue 6, vol. 5, June 2008, pp. 869-879.

[2] E. Milkova, “Algorithms: The Base of Programming Skills”, in *Proc. 29th International Conferences on Information Technology Inter-Faces*, University of Zagreb, Croatia, 2007, pp. 765 – 770.

[3] P. Prazak, “Recursively Defined Sequences and CAS”, in *Proc. 6th WSEAS/IASME International Conference on Educational Technologies (EDUTE’10)*, WSEAS Press, Kantoui, Sousse, Tunisia, 2010, pp. 58–61.

[4] S. Hubalovsky, E. Milkova, “Modelling of a real situation as a method of the algorithmic thinking development”, in *Proc. 6th WSEAS/IASME International Conference on Educational Technologies (EDUTE’10)*, WSEAS Press, Kantoui, Sousse, Tunisia, 2010, pp. 68–72.

[5] W. Dettmer, *The Logical Thinking Process: A Systems Approach to Complex Problem Solving*, Visconsin: Quality Press, 2007.

[6] M. Musílek, *Cipher and codes* [online]. 2010. Available: <http://www.musilek.eu/michal/sifry.html?menu=mat>.

[7] S. Singh, *The Code Book* [CD-ROM]. 2010. Available: <http://www.simonsingh.net>.

[8] *Letter Frequency*. Available: <http://en.wikipedia.org/wiki/Letter_frequency >.

[9] T. Jacobsen, “A Fast Method for the Cryptoanalysis of Substitution Ciphers”, *Cryptologia*, Vol. 19, 1995, pp. 265-274.

Stepan Hubalovsky was born in Trutnov, Czech Republic in 1970, he obtained master degree in education of mathematics, physics and computer science in 1995 and doctor degree in theory of education in physics in 1998 both in Faculty of Mathematics and Physics, Charles university in Prague, Czech Republic.
He worked 5 years as master of mathematics, physics and computer science on several secondary schools. He works as assistant professor on University of Hradec Kralove from 2006. He interested in algorithm development, programming and computer modelling.
RNDr. Stepan Hubalovsky, Ph.D. is member of Union of Czech Mathematicians and Physicist.

Michal Musilek was born in Pardubice Czech Republic in 1963, he obtained master degree in education of mathematics and physics in 1988, further approbation for education of computer science in 1993 and doctor degree in theory of education in physics in 2009 in University of Hradec Kralove, Czech Republic.
He worked 20 years as master of mathematics, physics and computer science on several secondary schools, including 8 years in the position headmaster of secondary school. He works as assistant professor on University of Hradec Kralove from 2009. He interested in theory of education in computer science, history of computers and cryptology.
PhDr. Michal Musilek, Ph.D. is member of Union of Czech Mathematicians and Physicist, its Mathematics Teachers Society.