Introduction to Exercise 1

Symmetrical cryptography – one of the oldest and simplest system of encrypting and decrypting of information where the same key is used in both directions. The security of the system is based on the secrecy of the key, which requires a secure way to reconcile the private key between the sender and the receiver.

The symmetric cryptography system is divided into two basic types: 1) substitution and 2) transposition ciphers.

Substitution cipher (in classical version) assumes that each character of the plain text is replaced by another character. The recipient reverses the substitution and obtains the plain text.

There are at least four kinds of substitution cipher:

a) Simple substitution cipher – a single character is replaced by single character taken from the text published in newspaper, book etc.
b) Homomophonic substitution cipher – similar to classical substitution cipher, but single character may have different assigned characters, for instance: letter A may be assigned to 5, 13, 25
c) Polygram substitution cipher – it encodes a group of characters, for instance ABA may be assigned to `RTQ`, `ABB` to `SLL` and so on.
d) Polyalphabetic substitution cipher – is based on idea of superposition of many classical substitution ciphers in turn.

Ceasar cipher – used by famous Roman impereur to communicate with the army is the axample of classical substitution cipher. Every letter in the plain text was replaced by the letter shifted three position (key = 3) to the right. So the command: `ATTACK AT DOWN` is encrypted to `DWWDFN DW GRZQ` according to the algorithm below:

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

Decryption is easy and requires to shift all letters of k = 3 position backward.

The simplicity of Caesar's cipher has brought him popularity today, as it still exists in the distributed operating system UNIX as a ROT13 program with the key = +13, so ROT13(`A`) = `N` for example. It means also that `M` = ROT13(ROT13(`M`)) – as `M` lays in the middle of the alphabeth length.

This cipher is not used for encryption of files, it is rather used in Usenet to hide abusive words, solutions of crosswords, puzzles and rebuses.

Exercise 1. The Caesar cipher

The purpose of this exercise is to demonstrate the universal form of Caesar's cipher described in VBA (for Excel) as shown in Figure 1. This version enables you to shift letters of any key value within the loop of 26 letters of English alphabet.

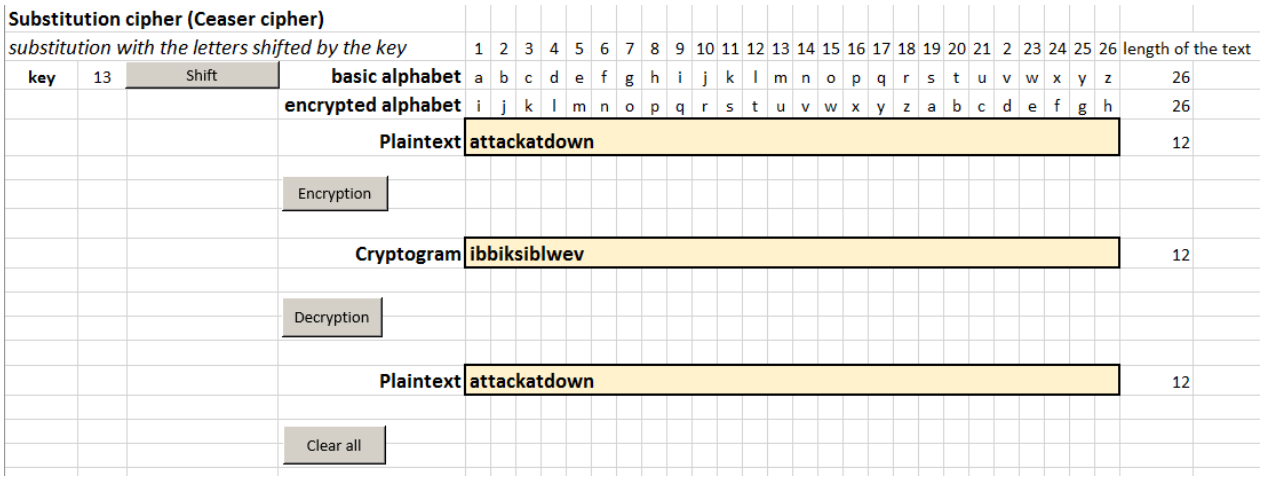


Fig.1. An Excel sheet illustrating the idea of Caesar's cipher

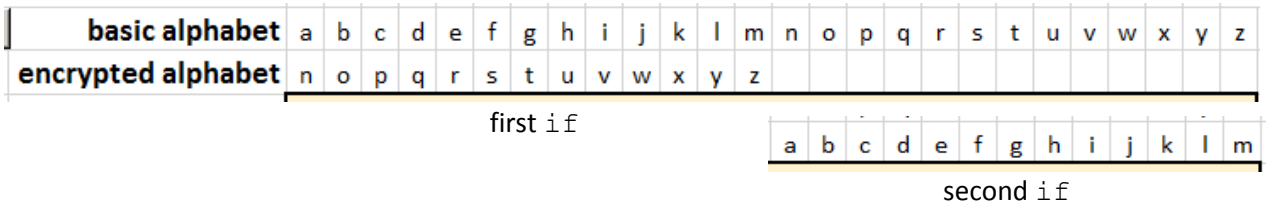
Encryption

At the beginning, the encrypted alphabet will be generated. It is describe in VBA code as following subroutine:

```
Sub encrypted_alphabet()
shift = Cells(3, 2)
For i = 97 To 122
 If i + shift <= 122 Then
  alphabet_enc(i - 97) = Chr(i + shift)
 End If
 If i + shift > 122 Then
  alphabet_enc(i - 97) = Chr(i - 26 + shift)
 End If
  Cells(4, 4 + i - 96) = alphabet_enc(i - 97)
Next i
End Sub
```

The main loop works in the range from 97 to 122 what corresponds to the characters from a to z in standard ASCII code. The variable `shift` determines the step of shifting with respect to "normal" alphabet.

First instruction `if` fills the `alphabet_enc` vector by characters, starting from the position 97 + `shift`. For instance, if `shift` = 8, then `alphabet_enc` vector will start from the 97 + 8 = 105 of ASCII character. The second instruction `if` fills the rest of vector by characters starting from letter a (97 in ASCII).



The loop `For` with `i` variable does the letter substitution one by one along the whole alphabet. Note the necessity of using a conditional functions to not go beyond the ASCII range.

Now, having already encrypted alphabet, is time to encrypt any plaintext by use of the following macro in EXCEL.

```
Sub Ceasar()
Erase cryptogram
For k = 97 To 122 ‘ generation of ”normal” alphabet
                  ‘ this is the beginning of comment in VBA
 alfabet(k - 96) = Chr(k)
Next k
a = div_text(Cells(5, 5), Cells(5, 31))
For j = 1 To Cells(5, 31)
For i = 0 To 25
 If txt_out(j) = alphabet(i + 1) Then
 cryptogram(j) = alphabet_enc(i)
 Cells(9, 4 + j) = cryptogram (j)
 i = 25
End If
Next i
Next j
Cells(9, 5) = ""
For i = 1 To Cells(5, 31)
 Cells(9, 5) = Cells(9, 5) & cryptogram(i)
Next i
End Sub
```

You can notice that macro `Caesar` begins with the generation of “normal “ alphabet. The next step is devoted to division of the plaintext into single characters, what enables performing of character’s substitution within two loops controlled by `j` and variables. The range of j loop depends on the length of the plaintext, while the `i` loop has constant range, depending on the number of characters in chosen alphabet. The conditional instruction `if` within both loops is responsible for placing a proper character from the encrypted alphabet into the vector called `cryptogram`.

Decryption

The process of decryption is nothing more then inversion of encryption. It is presented as the following subroutine in VBA code.

```
Sub de_Caesar()
Erase plaintext
Erase cryptogram
For k = 97 To 122 ' “normal” alphabet
 alphabet(k - 96) = Chr(k)
Next k
a = div_text(Cells(9, 5), Cells(9, 31))
For j = 1 To Cells(9, 31)
For i = 0 To 25
 If txt_out(j) = alphabet_enc(i) Then
  public(j) = alphabet(i + 1)
  i = 25
 End If
Next i
Next j
Cells(14, 5) = ""
For i = 1 To Cells(9, 31)
 Cells(14, 5) = Cells(14, 5) & public(i)
Next i
End Sub
```

The `div_text(…)` used above is a typical VBA function with some input parameters (references to cells addresses). Here you have its contents

```
Function div_text(txt_in As String, dl As Integer) As String
div_text = ""
For i = 1 To dl
  txt_out(i) = Mid(txt_in, i, 1)
Next i
  div_text = txt_out(dl)
End Function
```

Line `txt_out(i) = Mid(txt_in, i, 1)` returns a string containing a specified number of characters from an input string `text_in()`. Line `div_text = txt_out(dl)` is the assignment of any dummy value (in this case the length of the string) in order to be correct with formal requirements, but more important is filling the global vector `txt_out` by characters taken from another text string `txt_in`.

Macro buttons and their functionality

There are four buttons shown in Fig. 1 associated with Excel macros that are used for performing some action.

Shift	creates the encrypted alphabet with shifted characters, subroutine `alphabet_enc`
Encryption	activation of encryption subroutine `Caesar`
Decryption	activation of decryption subroutine `de_Caesar`
Clear all	clears the cells of encrypted alphabet, plaintext and cryptogram

All of them are chosen as the Button formant from Developer -> Insert Formants set of tools. They are "drawn" by mouse as the rectangle of desired size and can be easily moved to any place of the sheet.

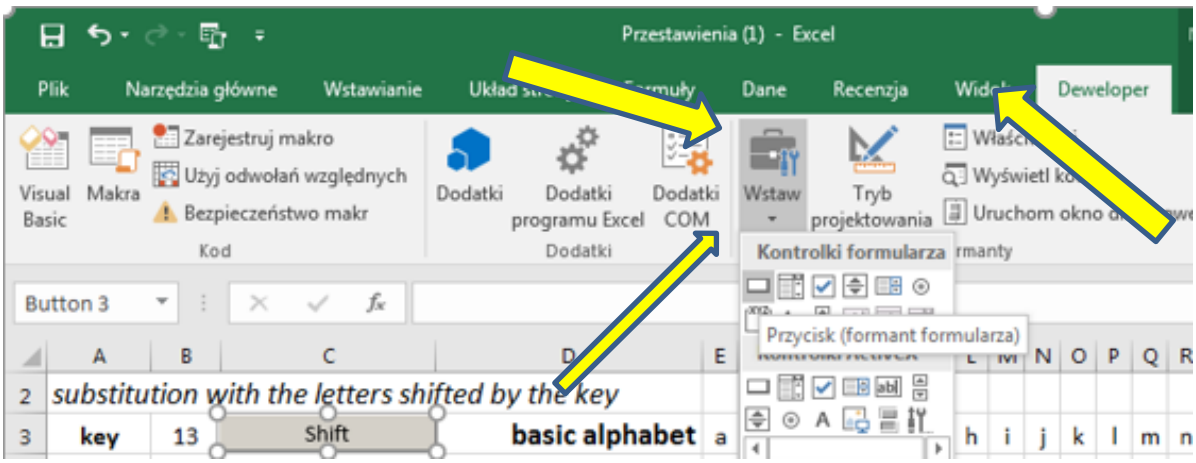


Fig.2. The Button's formant location

Each embedded button must be assign to the macro code by clicking twice the right mouse button and choosing proper macro name from the whole list of macros.

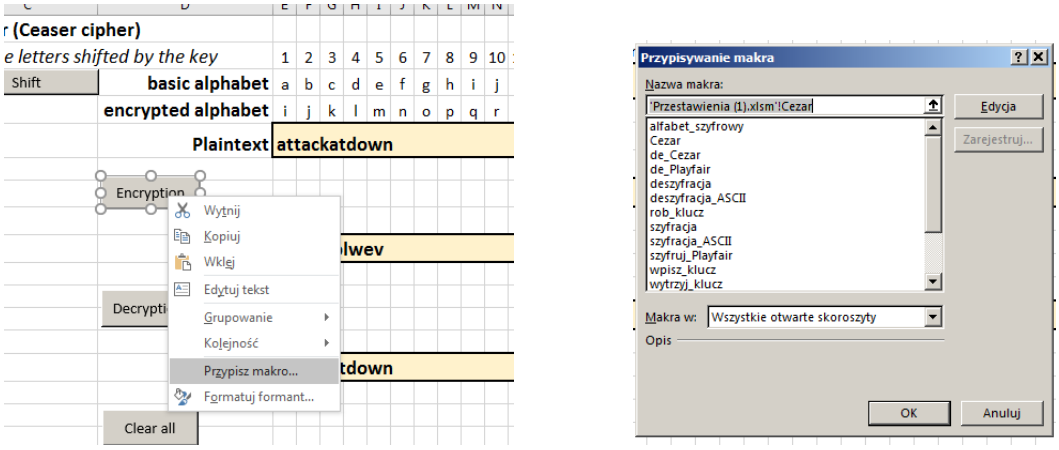


Fig. 3. The list of macros to be chosen while assignment of the Button

Here you have the listing of all functions and subroutines, including some declaration of variables that will be used in your VBA program.

```
Dim alphabet_enc(26), publ(500), alphabet(26), cryptogram(500)
Dim txt_out(500)
```

Global variables declarations

```
Function div_text(txt_in As String, dl As Integer) As String
div_text = ""
For i = 1 To dl
  txt_out(i) = Mid(txt_in, i, 1)
Next i
  div_text = txt_out(dl)
End Function
```

Declaration of function (must proceed the first call)

```
Sub alphabet_ciph()
shift = Cells(3, 2)
For i = 97 To 122
 If i + shift <= 122 Then
  alphabet_enc(i - 97) = Chr(i + shift)
 End If
 If i + shift > 122 Then
  alphabet_enc(i - 97) = Chr(i - 26 + shift)
 End If
  Cells(4, 4 + i - 96) = alphabet_enc(i - 97)
Next i
End Sub
```

Assigned to "Shift" macro

```
Sub Caesar()
Erase cryptogram
For k = 97 To 122 ' "normal" (public) alphabet
 alphabet(k - 96) = Chr(k)
Next k
a = div_text(Cells(5, 5), Cells(5, 31))
For j = 1 To Cells(5, 31)
For i = 0 To 25
 If txt_out(j) = alphabet(i + 1) Then
 cryptogram(j) = alphabet_enc(i)
 Cells(9, 4 + j) = cryptogram (j)
 i = 25
End If
```

Assigned to "Caesar" macro

```
Next i
Next j
Cells(9, 5) = ""
For i = 1 To Cells(5, 31)
 Cells(9, 5) = Cells(9, 5) & cryptogram(i)
Next i
End Sub
```

Assigned to "Caesar" macro

```
Sub de_Caesar()
Erase public
Erase cryptogram
For k = 97 To 122 '"normal" (public) alphabet
 alphabet(k - 96) = Chr(k)
Next k
a = div_text(Cells(9, 5), Cells(9, 31))
For j = 1 To Cells(9, 31)
For i = 0 To 25
 If txt_out(j) = alphabet_enc(i) Then
 publ(j) = alphabet(i + 1)
 i = 25
End If
Next i
Next j
Cells(14, 5) = ""
For i = 1 To Cells(9, 31)
 Cells(14, 5) = Cells(14, 5) & publ(i)
Next i
End Sub
```

Assigned to „de_Caesar" macro

```
Sub clear_all()
 Cells(5, 5) = ""
 Cells(9, 5) = ""
 Cells(14, 5) = ""
End Sub
```

Assigned to „Clear all" macro

Tasks to be performed

1. Edit VBA program for encryption of the plaintext by Caesar cipher.
2. Try to encrypt the message to your PC lab's neighbour and ask him to find a plaintext from the cryptogram.
3. Do the same with your neighbour's cryptogram.
4. Analyse the easiest way of getting the plaintext from cryptogram
5. Estimate the level of difficulty to break the cipher.
6. Write down the list of weak points of Caesar ciphering.