

# Assignment 2 – Substitution-based Encryption

**Title:** Assignment 2 - Substitution-based Encryption

## Step 1 - Problem Statement:

Write a program that encrypts and decrypts messages (strings) using the Substitution algorithm called **ROT13** (see [Wiki](#) for a description of this algorithm).

Your program must keep instructing the user to ...

Please, enter a message to encrypt/decrypt or enter an empty string to exit the program:

until the user enters an empty string. When this occurs, your program must terminate.

If the user enters a message (non-empty string), your program must encrypt this message, then decrypt its encrypted version and display the original message, the encrypted version as well as the decrypted version on the computer monitor screen.

## Requirements:

- You must create your own Python 3 program using IDLE. You cannot use AI tools to create your program nor can you use someone else's program. For more details regarding Academic Honesty (or lack of) and what is permissible and what is not, please, read [A word about Academic Honesty](#) under **Lecture 1** on our course web site.
- Your program must start when the menu option **Run** is selected or **F5** is pressed once we have opened your program file in IDLE.
- You must use the **ROT13** Substitution algorithm as it is described by Wikipedia. No other descriptions of this algorithm are acceptable.
- When implementing your program, you cannot use the Python implementations listed on this Wikipedia web site (and on any other web site for that matter). This is to say that ...
  - You cannot use the module **codecs** and its functions such as **encode(...)**.
  - You cannot use a **list comprehension**. (We shall cover list comprehensions later on this semester.)
  - You cannot use the functions **maketrans(...)**, **translate(...)**, **join(...)** and **lambda**.
  - You must write your Python program on your own. **Warning:** If your program replicates any of the implementations found on Wikipedia or on any other web sites, you will get 0 for this assignment. 😞
- If you use a **while** loop in your program, your while loop must be composed of a Boolean condition (i.e., Boolean expression) which must evaluate to **True** and eventually to

**False** depending on the value assigned to the variable(s) contained in this Boolean expression. You cannot use a **while True:** and **break** or **exit( )** as part of the body of your loop.

- You can also use **for** loops.
  - Remember, **while** loops are great when you do not know how many times your code needs to iterate, and **for** loop are great when you do know how many times your code needs to iterate.
- Your program must have a header comment block containing the expected sections (4 sections) and located at the expected place in your program.
- Your program must follow these Good Programming Style (GPS) rules:
  - Variables must be named in a descriptive fashion (describing their purpose).
  - All import statements (if any) must be placed at the top of the file, immediately after the header comment block.
  - Each line of your code must not exceed 80 characters in length.
    - Why? This is to help you develop a habit of *good code formatting*.
    - In most cases, you can break long lines of code (statements) into separate steps or have them span multiple lines. You could also use intermediate variables to store parts of long messages to be printed.
    - How to know the length of your lines (code and comments) when you are using the text editor of IDLE? Answer: Place the cursor by the last (right-most) character of a line and look at the Col: (column) figure at the bottom right-hand side corner of the text editor window.
- Save your program naming it **Assignment2.py**

You are encouraged to expand on this assignment by using Python statements we have not seen in class in order to expand your understanding of Python, **as long as** you satisfy all the **Requirements** stated in this assignment. There are no extra points for this, but it will help you to better learn Python and have fun!

## Step 2 - Design:

Before you start coding, make sure you design your algorithm using English (natural language) or using pseudocode. You must include your algorithm as comments in your program.

## Step 3 – Implementation:

Once you have included your algorithm as comments in your empty program, translate each of the steps of your algorithm (each comment) into Python code. You must leave the comments in the code.

**Incremental development:** Implement a few comments at a time, then ...

## Step 4 - Testing:

... test as you go! Avoid the painful **Big Bang approach** to the software development process. 😞

**Incremental testing:** Test as you go! This means that as you are translating your comments (algorithm steps) into Python statements, make sure your program executes as expected.

### Submission:

- Submit your program on CourSys (<https://coursys.sfu.ca/2024sp-cmpt-120-d3/>). Click on the course activity called **Assignment 2**, then click on the option **Make Submission** on the left and finally, follow the instruction to browse for your program file.
- **Note** that you can submit your program as often as you wish. As long as your submissions are done before or on the due date and time, your assignment will be marked. CourSys will not stop you from submitting your program late, i.e., after the due date and time, but if your program is late, it will receive 0 marks.

### How your Assignment 2 will be marked:

- When the TA marks your Assignment 2, he will be looking at
  - whether your program solves the problem using a few test cases. For example,
    1. a message containing upper case letters,
    2. a message containing lower case letters,
    3. a message containing a mix of upper and lower case letters,
    4. a message containing symbols (&, \$, etc.),
    5. a message containing a mix of letters (upper and/or lower) and symbols,
    6. a message containing numbers,
    7. a message containing a mix of numbers and letters (upper and/or lower),
    8. a message containing a mix of letters (upper and/or lower), numbers and symbols.
  - whether your program satisfies **all the requirements** stated in this assignment.
- The rubric for Assignment 2 is based on the above. Make sure your program satisfies the above before submitting it on CourSys.

If you have any questions, drop by our office hours or post them on our Discussion Forum.  
**Suggestion: Make sure you deal with your questions as you go. Do not wait until exam time.**

**Finally, there are no extension granted unless for medical reason once the [Official Medical Certificate](#) has been completed and submitted to the instructor.**