Sabancı University

Faculty of Engineering and Natural Sciences

CS204 Advanced Programming

Fall 2019-2020

Homework 6

Templates and Bit Operations

Due: **22/11/2019, 11:55pm**

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You HAVE TO write down the code on your own.

You CANNOT HELP any friend while coding.

Plagiarism will not be tolerated!

Background

Cryptography, in which anything digital heavily depends on for (to-some-degree) secure data sharing, is an important part of the computer science, if not tedious. In this homework, you are going to perform a very simple cryptography operation. Any cryptologic algorithm performs its task by **encrypting** the message (making it unreadable) and **decrypting** it (converting back to human readable form). Most of the cryptography algorithms use a **key** to encrypt and decrypt messages.

In this homework, you are given a cpp file with main. You are going to write a template class which operates encryption and decryption operations on **chars, short integers, long integers, long long integers**. The encryption operation will consist of three stages:

1- **XOR** operation
2- Circular shifting
3- Flipping the given bits

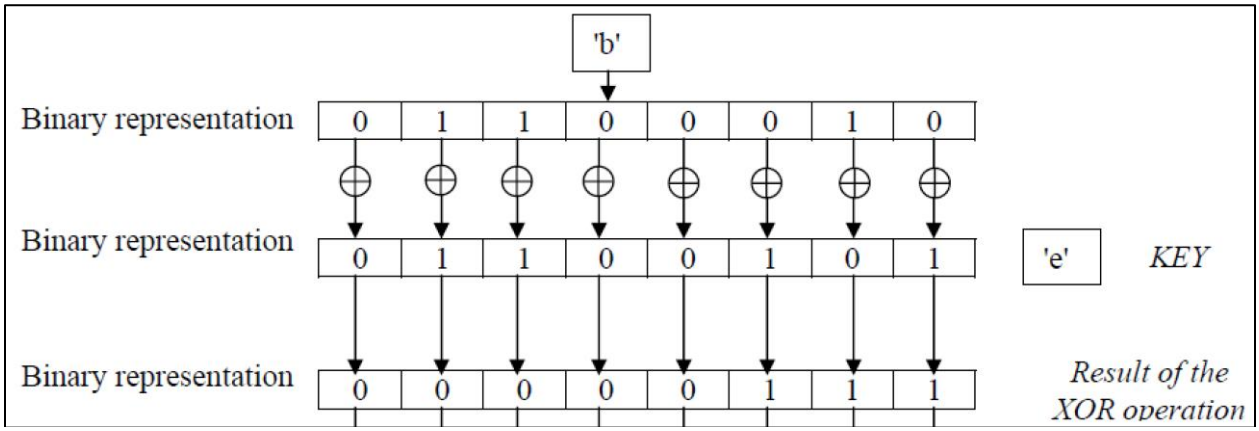
Your template class has 3 fields: **key** for XOR operation, **nShift** for number of shifts in circular shifting and **listFlips** for the list of bits to flip.

The class, named **Encrypter,** will have a constructor and 2 member functions: **encrypt, decrypt.** Encrypt function will do the afore-mentioned operation in the given order. And decrypted function will do the operations in the reverse order to decrypt the data successfully.

1- XOR operation

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

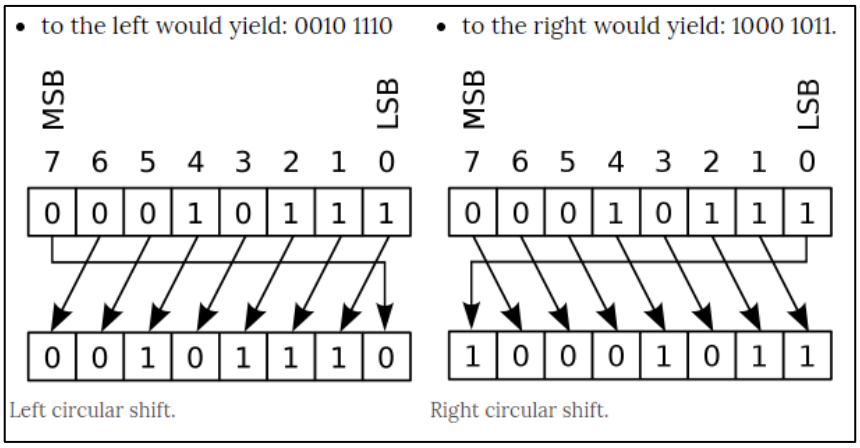
It will take the input and operate bitwise xor operation on it with the **key.** The following image shows the result of sample XOR operation.



Then, the output of the xor operation will be given to **Circular shifting** operation as input.

2- Circular Shifting

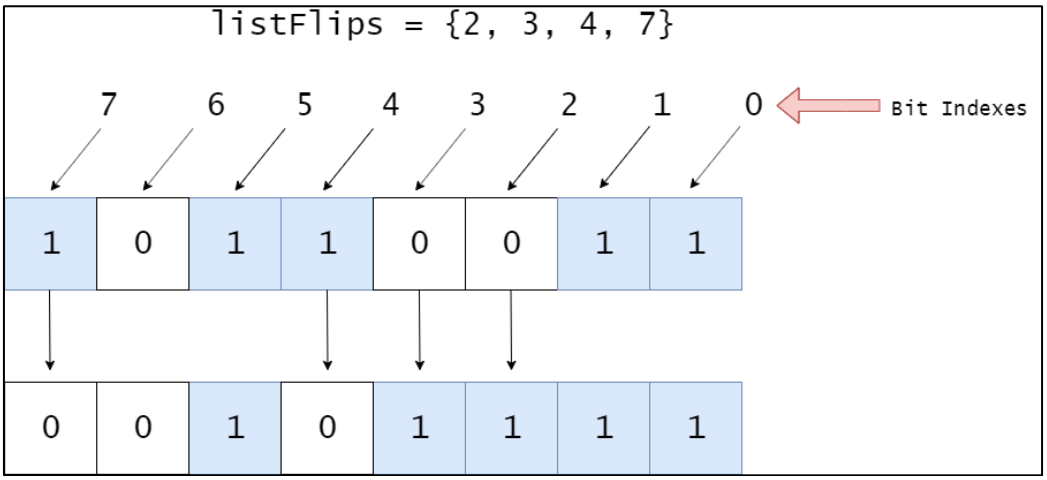
It is the shifting operation with moving the final bit to the first bit while shifting. With this approach of shifting, we achieve to prevent loss while shifting. Let's say we want to shift our 0001 0111 with circular shift operation to left and right. Images below shows the operation.



In the **encryption** function you are going to implement, it will shift the bits **nShifts** many times to **left**. And at **decryption** stage, it will shift the bits **nShifts** many times to **right**. If **nShifts** is greater than or equal to the bit size of data to be shifted, it will take the mod(%) of nShifts and then shift that many times.

3- Flip the given bits

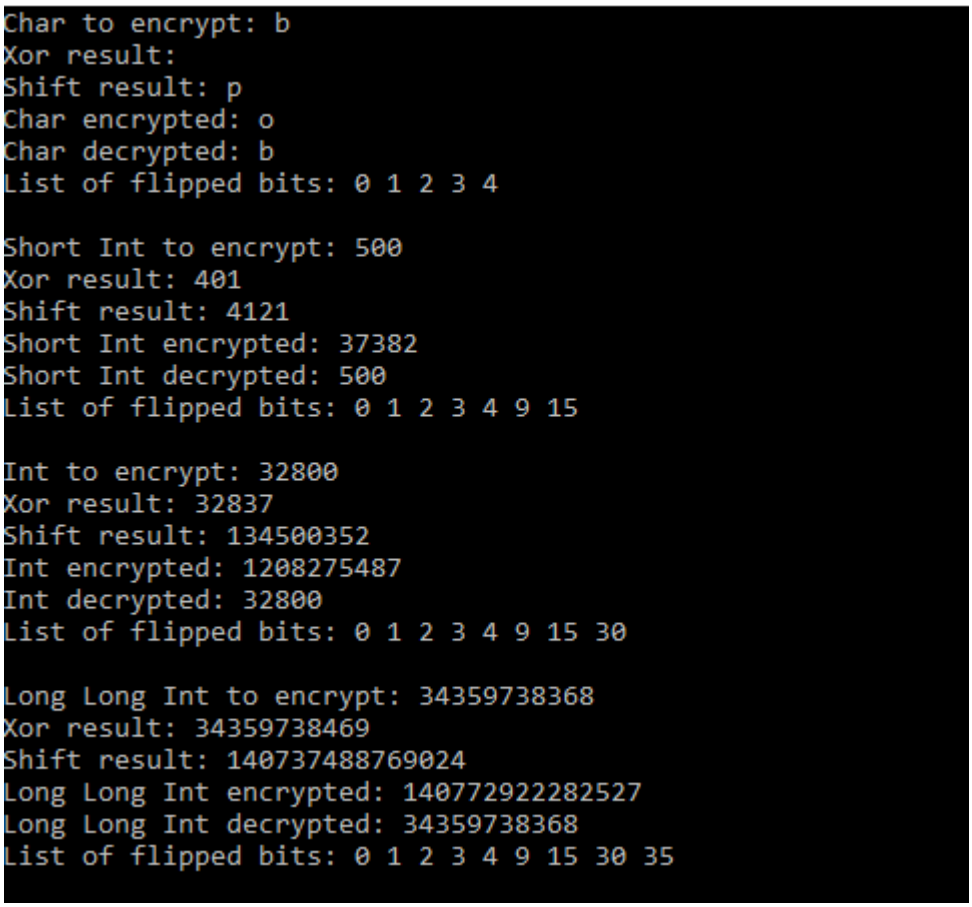
While creating Encrypter object, a vector of bits to flip will be given. Let's say **listFlips** is given as follows: {2, 3, 4, 7}. Then as seen in the following image, 2nd, 3rd, 4th and 7th bits of the result of circular shift operation will be flipped.



If there is a bit greater than the bit count of the data type, they will be neglected and won't be in the listFlips vector. Let's say our datatype to be encrypted and decrypted is char which is 1 byte(8 bits). If the list of bits to flip given to the constructor of the Encrypter for char is given as follows: {0, 5, 7, 12, 19}. Since 12 and 19 is greater than 7(the maximum index we can hav efor char datatype), they will be rejected and our **listFlips** vector will be as follows: {0, 5, 7}. Than only these bits will be flipped.

Output

You are given a cpp file which includes main function. You don't need to modify main function, you also don't need to have additional cpp/header files. The needed output for main function given to you is as follows.



Some Important Rules

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homeworks we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in *Release* mode and **we may test your programs with very large test cases**.

What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.
Name your cpp file that contains your program as follows:

"SUCourseUserName_YourLastname_YourName_HWnumber.cpp"

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is valent, name is Valentina, and last name is Tereşkova, then the file name must be:

Valent_Tereskova_Valentina_hw1.cpp

Do not add any other character or phrase to the file name. Make sure that this file is the latest version of your homework program. Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains your cpp file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

SUCourseUserName_YourLastname_YourName_HWnumber.zip

For example zubzipler_Zipleroglu_Zubeyir_hw1.zip is a valid name, but

hw1_hoz_HasanOz.zip, HasanOzHoz.zip

are **NOT** valid names.

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
CS204 Team (M. Yusa Erguven, Kamer Kaya)