C++ Programming Sections 1782 & 4075

Deadline March 16, 2021 by 9:00 AM PST

Programming Assignment 2 (100 Points):

Simple Cipher: C++ Strings, References & Functions (updated March 6, 2021)

Assignment Introduction:

Write a program to encrypt and decrypt a plaintext (p) string of lower-case letters, the digits 0-9 and two special characters , **#,!**. The plaintext will not contain spaces or other characters (e.g.,etc.). The program will have at least two user-defined functions **encrypt** and **decrypt**.

Your program will accept command line arguments to allow the user to enter in a key value k, a plaintext string of characters just described $(-p)$ or a ciphertext $(-C)$ and exactly one of the options to encrypt $(-E)$ or decrypt $(-D)$. Refer to the example usage on page 3 for how to run the program using the command line arguments and the skeleton code on page 4 for a general outline of the code.

Program Description and Functionality: Provide the following in your source file:

1. Comments at the top of your **.cpp** file.
2. A main function with command line arguments `main(int argc, char* argv[])`
3. **Use only C++ headers** that place all standard-library routines in `namespace std`. For C++ strings use `#include <string>`; for standard input and output use `#include <iostream>` to access `std::cout;std::cin`; and use `#include <cstring>` for C-style string functions like `std::strcmp` and `std::strlen` and if needed `#include<cstdlib>` for helper functions like `std::atoi`, etc.
4. The use of command line arguments (int argc, char* argv[]) to get the plaintext (-p SoMetEXt) or cipher text (-C pKjAPbuP), the shift amount $-k$ 7 and the options to encrypt (-E) or decrypt (-D).
5. The program needs to work for upper-case & lower-case letters, digits 0-9, & the symbols # and . Refer to the declaration of **codebook** below for the supported character-set:

```
char codebook[] =
{ 'Z','z','Y','y','X','x','W','w','V','v','U','u','T','t','S','s','R','r','Q','q',\
  'P','p','O','o','N','n','M','m','L','l','K','k','J','j','I','i','H','h','G','g',\
  'F','f','E','e','D','d','C','c','B','b','A','a','9','8','7','6','5','4','3','2',\
  '1','0','#','!'
};
```

6. Write at least two functions, `encrypt()` & `decrypt()` with the following function prototypes:

```
// plaintext and k are the values parsed from the command line
void encrypt(std::string& plaintext, int k);

// ciphertext and k are the values parsed from the command line
void decrypt(std::string& ciphertext, int k);
```

7. To encrypt a plaintext string use the following formula to compute an index i into the `codebook` with length `#define CBL 64 //Code Book Length`:

$$C_i = E(p) = (p + k) \mod CBL,$$

where *mod* is the modulus operator %, C is the ciphertext, E is the encryption function, p is the index of the plaintext character, k is the key shift, and CBL is the length of the `codebook` array.

8. To decrypt a ciphertext string use the following formula to compute an index i into the `codebook` array:

$$p_i = D(C) = (C - k) \mod CBL, \qquad (C - k) >= 0$$
$$p_i = D(C) = (((C - k) \mod CBL) + CBL) \mod CBL, \qquad (C - k) < 0$$

where *mod* is the modulus operator %, C is the index of the ciphertext character, D is the decryption function, p is the plaintext, k is the key shift, and CBL is the length of the `codebook` array.

In C & C++, mod, or remainder of division, is the % symbol. For negative $C - k$ numbers refer to the references on page 4 for more details about the modulus operator in C++ with negative numbers.

Where to do the assignment

You can do this assignment on your own computer, or using the SMC Virtual labs with Citrix. In either case, ensure the code compiles and runs on Windows. Submit one **.cpp** file named **A02.cpp**. Do not use any other name or else points will be deducted.

Submitting the Assignment

Include your name, your student id, the assignment number, the submission date, and a program description in comments at the top of your files, and use the CS52 Programming Guide for coding style guidance. Submit the assignment on Canvas (`https://online.smc.edu`) by **uploading your .cpp file** to the Assignment 2 entry as an attachment. Do not cut-and-paste your program into a text window. Do not hand in a screenshot of your program's output. Do not hand in a text file containing the output of your program.

Saving your work

Save your work often on a flash-drive or to the cloud (e.g., GoogleDrive, Microsoft OneDrive, Canvas, etc.). Always save a personal copy of your files (e.g. .cpp, .h, etc.). Do not store files on the virtual lab computers.

References:

- Microsoft C++ Language Reference
- ISO C++ Standard Library and other refs
- *'The modulus operator yields the remainder given by the following expression, where e is the first operand and $e2$ is the second: $e1 - (e1/e2) * e2$, where both operands are of integral types.'* Microsoft C++ Modulus & Multiplication Operators

Example usage:

```
>A02.exe -p #Bc!! -k 3 -E           //encrypt the string #Bc!!
The plaintext was: #Bc!!
The ciphertext is: zaAYY
enter any key to exit


>A02.exe -C zaAYY -k 3 -D           //decrypt the string zaAYY
The ciphertext was: zaAYY
The plaintext is: #Bc!!
enter any key to exit


>A02.exe -k 5 -p !01#z7!9# -E       //encrypt the string !01#z7!9#
The plaintext was: !01#z7!9#
The ciphertext is: XYzyW2X4y
enter any key to exit


>A02.exe -C XYzyW2X4y -k 5 -D       //decrypt the string XYzyW2X4y
The ciphertext was: XYzyW2X4y
The plaintext is: !01#z7!9#
enter any key to exit


>A02.exe -p sometext -k 7 -E        //encrypt the string sometext
The plaintext was: sometext
The ciphertext is: OKIAPATP
enter any key to exit


>A02.exe -C OKIAPATP -k 7 -D        //decrypt the string OKIAPATP
The ciphertext was: OKIAPATP
The plaintext is: sometext
enter any key to exit


>A02.exe -k 9 -E -p tH!s!sv3ryYc00l   //encrypt the string tH!s!sv3ryYc00l
The plaintext was: tH!s!sv3ryYc00l
The ciphertext is: OdVNVNQyMTu5WWG
enter any key to exit


>A02.exe -C OdVNVNQyMTu5WWG -k 9 -D   //decrypt the string OdVNVNQyMTu5WWG
The ciphertext was: OdVNVNQyMTu5WWG
The plaintext is: tH!s!sv3ryYc00l
enter any key to exit
```

Skeleton code

```
// Name: Your First Name & Last Name
// SSID: Student ID Number
// Assignment #: 2
// Submission Date: 3/16/2021
//
// Description: A program to encrypt and decrypt a message using a shift cipher
// The plaintext message must only contain lowercase alpha and digits 0-9
//
// command line arguments
// -p ThePla!Nt#xTM#ss2ge1 - the plaintext (p) message to be encrypted
// -C pC9lG1VjOwSpiwNNXB9x - the cipher (C) text to be decrypted
// -k 9                    - the key (k) shift value
// -E                      - the option to encrypt (E)
// -D                      - the option to decrypt (D)
//

#include <iostream> //std::cout, std::cin, etc.
#include <cstdlib>  //C++ version of stdlib.h
#include <cstring>  //C++ version of string.h
#include <string>

// Todo A2: encrypt using std::string
void encrypt(std::string& plaintext, int k);

// Todo A2: decrypt using std::string
void decrypt(std::string& ciphertext, int k);

int main(int argc, char *argv[])
{
        // Example variables for A02
        int k = 0;                // k shift
        int msg_index = 0;        // msg_index of plaintext or ciphertext in argv
        std::string msg;          // string to hold the plaintext or ciphertext;
        bool do_encrypt = false;  // True for encrypt or False for decrypt

        // Todo A2: process the command line arguments
        for (int i = 1; i < argc; i++){
                // use strcmp to compare command line switches to argv[i]
                // ....your code goes here
        }

        //
        // Todo A2: call encrypt or decrypt functions
        //
        if (do_encrypt)         //encrypt plaintext
                encrypt(msg, k);
        else
                decrypt(msg, k);//decrypt ciphertext

        //
        // Todo A2: print the original message and the plaintext | ciphertext
        //

        return 0;
}//main

//Todo A2: function definitions go here
```