CSCE 1030: Homework 4

Due: 11:59 PM on Thursday, November 2, 2017

BACKGROUND INFORMATION:

A cryptosystem is a system for encryption and decryption. Encryption, also called encoding or enciphering, involves changing the original message into a secret message, while decryption, also called decoding or deciphering, involves just the opposite – changing the secret message back into its original readable form. In cryptography, a *plaintext* file is a file containing data in its original, or readable, form while a *ciphertext* file is a file containing data in coded, or scrambled, form.

In this assignment, you will be creating a modified stream cipher that shifts each original letter or digit to a new, encrypted letter or digit, respectively, corresponding to individual, randomly generated integral key. For example, consider the plaintext to be encrypted is: "`ABC 1.`", without the quotes. Then, for each letter and digit, we generate a random integer between 3 and 277 that serves as the key for the corresponding letters and digits. Consider the following example:

```
Plaintext:      ABC 1.
Key:            76 97 74 75
Ciphertext:     YUY6.
```

Thus, the `A` would be shifted 76 places in the alphabet, which means that it "rolls over" twice, and then gets shifted 24 places in the alphabet to the ciphertext `Y`. Notice that there are only four keys generated for the six characters. Whitespace (and any unsupported characters) should be discarded, so no key is generated in this case. Note that the whitespace has been removed in the ciphertext. Additionally, punctuation should be kept as is, meaning that it is not discarded, but left in its original form (e.g., the "`.`" in the above example remains a "`.`") in the ciphertext.

Mathematically, each letter of the plaintext and key can be given a number and a formula can be derived to encrypt for `c` (i.e., ciphertext) and `m` (i.e., plaintext) using `k` (i.e., key) as follows:

$$c = (m + k) \% 26$$

$$m = (26 + c - k) \% 26$$

Note that formula assumes `A` is 0, `B` is 1, and so forth with nothing to distinguish between uppercase `A` and lowercase `a`. Since you will most likely be using the character ASCII values, you will have to modify these formulas to fit your needs (e.g., the letter `A` has an ASCII value of 65), but this should give you a place to start. One other item to consider is that characters (i.e., letters and digits in this case) roll over. For example, in shifting the letter `Z`, then next letter would be the letter `A` (both uppercase).

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

CSCE 1030: Homework 4

PROGRAM DESCRIPTION:

The purpose of this programming project is to write a C++ program to implement a simple stream cipher that incorporates topics related to file I/O. In particular, this programming assignment will read an input file, encrypt or decrypt the file based on the requested operation, and write the results to an output file.

REQUIREMENTS:

- As with all programs in this course, your program's output should initially display the department and course number, your name, your EUID, and your e-mail address. This functionality will be implemented using a function that you call from your `main()` function.
- In your `main()` function, you will prompt the user whether he/she would like to encrypt or decrypt a file. If a valid response is not input, you are to repeatedly re-prompt the user until a valid response is input.
- Also inside your `main()` function, you will then prompt the user to enter the name of the input file to read from and the output file to write the resulting plaintext or ciphertext as appropriate. You will attempt to open both files in your `main()` function. If there is a problem opening the file, you will display a meaningful error and exit the program using the appropriate exit status. You may assume that the length of the file name does not exceed 32 characters (33 with the terminating null character).
- If there are no errors causing you to terminate the program at this point, you will then call the encryption or decryption function that will process the input file and write the results to an output file.
    - The files that you will encrypt or decrypt can contain uppercase alphabetic characters (`A` – `Z`), lowercase alphabetic characters (`a` – `z`), digits (`0` – `9`), punctuation, and whitespace, which may include any white space character such as a blank space, a tab, or a newline character.
    - The user-defined functions to process the input and output files (i.e., to encrypt and decrypt) should accept two parameters as a minimum, the input file stream and the output file stream, but it may utilize additional parameters as needed. You must process each file character-by-character (i.e., using the `get` and `put` member functions).
    - You will handle *encryption* in the following manner:
        a. You will prompt the user to enter the name of a different file that will contain your encryption keys. You will attempt to open the file in this function. If there is a problem opening the file, you will display a meaningful error and exit the program using the appropriate exit status. You may assume that the length of the file name does not

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

exceed 32 characters (33 with the terminating null character). Any randomly generated keys for alphabetic characters and digits will be written to this file (that will later be used in decryption). Be sure to close the key file when you are done.

b. You will encrypt all alphabetic characters, both uppercase and lowercase, as well as all digit characters using a randomly generated integral value between 3 and 277, inclusively. Be sure to seed your random number. Uppercase characters should encrypt to uppercase characters, lowercase characters to lowercase characters, and digits to digits. For example, if the key is `3`, `A` would be replaced with the `D`, `B` would be replaced by `E`, …, `Y` would be replaced by `B`, and `Z` would be replaced by `C`. Similarly, if the key is `3`, `a` would be replaced with the `d`, `b` would be replaced by `e`, …, `y` would be replaced by `b`, and `z` would be replaced by `c`. Additionally, if the key is `3`, `0` would be replaced by `3`, `1` would be replaced by `4`, …, `8` would be replaced by `1`, and `9` would be replaced by `2`.

c. If a punctuation character is read in, you will not encrypt the character, but simply keep the punctuation character as is and write it to the ciphertext file. This means that you will not generate a key for punctuation characters.

d. If a whitespace character is read in, such as a blank space, a tab, or a newline character, you will not encrypt the whitespace character, but simply discard and not write it to the ciphertext file. This means that you will not generate a key for whitespace characters.

e. Any other characters not included in the above description shall be ignored (i.e., discarded), but an error message will be displayed and encryption should continue.

f. All encrypted and punctuation characters based on these requirements will be written to the ciphertext file specified by the user.

2. You will handle *decryption* in the following manner:

a. You will prompt the user to enter the name of the file that contains your encryption keys. You will attempt to open the file in this function. If there is a problem opening the file, you will display a meaningful error and exit the program using the appropriate exit status. You may assume that the length of the file name does not exceed 32 characters (33 with the terminating null character). you will use this file to read the randomly generated keys for alphabetic

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

characters and digits in the encryption process. Be sure to close the key file when you are done.

b. You will decrypt all alphabetic characters, both uppercase and lowercase, as well as all digits, using the correct randomly generated key for each encrypted character. Uppercase characters should decrypt to uppercase characters, lowercase characters to lowercase characters, and digits to digits. For example, if the key is `3`, `D` would be replaced with the `A`, `E` would be replaced by `B`, …, `B` would be replaced by `Y`, and `C` would be replaced by `Z`. Similarly, if the key is `3`, `d` would be replaced with the `a`, `e` would be replaced by `b`, …, `b` would be replaced by `y`, and `c` would be replaced by `z`. And finally, if the key is `3`, `3` would be replaced by `0`, `4` would be replaced by `1`, …, `1` would be replaced by `8`, and `2` would be replaced by `9`. It is important that the key used to encrypt this file is the SAME key that is used to decrypt the file, or you will not get the expected results.

c. If the ciphertext file contains punctuation, you will do no decryption, but simply keep the punctuation character as is and write it to the plaintext file.

d. Any other characters not included in the above description shall be ignored (i.e., discarded), but an error message will be displayed and encryption should continue.

e. All decrypted and punctuation characters based on these requirements will be written to the plaintext file specified by the user.

- Be sure to close both the input and output file after you are done processing them.
- See the sample program output for examples of what is expected. You should contact your instructor if there is any question about what is being asked for.
- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code.
- Your program source code should be named "**`homework4.cpp`**", without the quotes.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., `cse01`, `cse02`, …, `cse06`), so you should make sure that your program compiles and runs on a CSE machine.

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

CSCE 1030: Homework 4

- **This is an individual programming assignment that must be the sole work of the individual student.**

You may assume that all input will be of the appropriate data type, although the value itself may not be valid.

You shall use techniques and concepts discussed in class – you are not to use global variables, `goto` statements, or other items specifically not recommended.

DESIGN (ALGORITHM):

On a piece of paper (or word processor), write down the algorithm, or sequence of steps, that you will use to solve the problem. You may think of this as a "recipe" for someone else to follow. Continue to refine your "recipe" until it is clear and deterministically solves the problem. Be sure to include the steps for prompting for input, performing calculations, and displaying output.

You should attempt to solve the problem by hand first (using a calculator as needed) to work out what the answer should be for a few sets of inputs. Calculations could include an actual formula for how the encryption/decryption is done and the expected results based on a key.

Type these steps and calculations into a document (i.e., Word, text, PDF) that will be submitted along with your source code. Note that if you do any work by hand, images (such as pictures) may be used, but they must be clear and easily readable. This document shall contain both the algorithm and any supporting hand-calculations you used in verifying your results.

SAMPLE OUTPUT (input shown in **bold green**):

```
$ more plaintext1
AbCdEfGhIjKlMnOpQr,StUvWxYz0123456789.
$ ./a.out
     +---------------------------------------------+
     |       Computer Science and Engineering      |
     |        CSCE 1030 - Computer Science I       |
     |   Student Name     EUID    euid@my.unt.edu  |
     +---------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? E
Enter the name of your input file you want to encrypt: plaintext1
Enter the name of the output file to write the ciphertext: ciphertext1
Enter the file name that will contain your encryption keys: key1
$ more ciphertext1
QpQtUeJjYoWxLsDyGw,MqIcLyMz6039891034.
$ more key1
120 66 118 224 120 155 3 132 146 213 220 272 207 31 171 113 250 31 98 205 14
33 249 235 40 182 196 219 181 166 184 244 5 253 85 5
$ ./a.out
     +---------------------------------------------+
```

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

CSCE 1030: Homework 4

```
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I         |
     |   Student Name     EUID    euid@my.unt.edu    |
     +-----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? D
Enter the name of your input file you want to decrypt: ciphertext1
Enter the name of the output file to write the plaintext: ptext1
Enter the file name that contains your encryption keys: key1
$ more ptext1
AbCdEfGhIjKlMnOpQr,StUvWxYz0123456789.
$ more plaintext2
This life, which had been the
tomb of his virtue and of his
honour, is but a walking
shadow; a poor player, that
struts and frets his hour upon
the stage, and then is heard
no more: it is a tale told by an
idiot, full of sound and fury,
signifying nothing.
     -- William Shakespeare
$ ./a.out
     +-----------------------------------------------+
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I         |
     |   Student Name     EUID    euid@my.unt.edu    |
     +-----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? e
Enter the name of your input file you want to encrypt: plaintext2
Enter the name of the output file to write the ciphertext: ciphertext2
Enter the file name that will contain your encryption keys: key2
$ more ciphertext2
Trlbhfre,qrzsvednxpxvdpaoqyavggktorjmsnhdjzkhpbqckpex,fgxamfggjcoozzbpsxg;zed
szzfhvpk,crvmqctagcxipfltjreydvuufekkwoaqqzbzc,zchegppbfkcugg
dbxpik:nbmfjcgznkqzupwlidgbyd,bxzmamtcnyvlrdyyia,vmfguayrraprrkoxu.--
MoabphxApjxiocmwiy
$ more key2
182 218 81 243 230 231 90 156 98 88 199 250 40 101 237 192 100 167 175 112 36
8 126 255 28 246 181 215 131 25 132 131 227 217 96 45 102 87
 59 276 188 11 83 78 241 243 269 66 75 235 244 188 101 40 230 214 19 31 114
240 50 18 84 235 19 163 254 15 249 217 62 259 93 15 56 34 218
228 189 153 245 201 35 114 47 175 128 191 132 136 13 270 75 177 64 156 228 15
16 129 127 68 271 66 84 234 92 114 203 126 9 99 45 220 180 5
8 183 253 232 233 41 186 167 77 219 80 253 143 55 232 202 15 3 146 91 37 157
69 214 187 138 82 91 113 113 188 248 191 69 54 222 121 144 25
8 37 125 151 159 71 218 218 126 263 248 53 194 163 27 196 71 63 18 115 238
104 227 82 69 262 133 56 51 71 272 177 104 243 160 72 80 263 24
 29 136 218 222 94 6 15 224 59 189 141 86 242 139 91 160 152 169 268 22 43
124
$ ./a.out
     +-----------------------------------------------+
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I         |
```

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

CSCE 1030: Homework 4

```
     |   Student Name     EUID    euid@my.unt.edu   |
     +----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? d
Enter the name of your input file you want to decrypt: ciphertext2
Enter the name of the output file to write the plaintext: ptext2
Enter the file name that contains your encryption keys: key2
$ more ptext2
Thislife,whichhadbeenthetombofhisvirtueandofhishonour,isbutawalkingshadow;apo
orplayer,thatstrutsandfretshishouruponthestage,andthenisheard
nomore:itisataletoldbyanidiot,fullofsoundandfury,signifyingnothing.--
WilliamShakespeare
$ more plaintext3
Mon Oct 23 20:45:27 CDT 2017
$ ./a.out
     +----------------------------------------------+
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I        |
     |   Student Name     EUID    euid@my.unt.edu   |
     +----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? G
Would you like to ENCRYPT or DECRYPT a file (E or D)? F
Would you like to ENCRYPT or DECRYPT a file (E or D)? e
Enter the name of your input file you want to encrypt: plaintext3
Enter the name of the output file to write the ciphertext: ciphertext3
Enter the file name that will contain your encryption keys: key3
$ more ciphertext3
FveYrk8158:86:85WYO4336
$ more key3
253 7 199 192 67 147 66 78 263 128 44 71 126 138 150 151 73 142 13 182 189
$ ./a.out
     +----------------------------------------------+
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I        |
     |   Student Name     EUID    euid@my.unt.edu   |
     +----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? d
Enter the name of your input file you want to decrypt: ciphertext3
Enter the name of the output file to write the plaintext: ptext3
Enter the file name that contains your encryption keys: key3
$ more ptext3
MonOct2320:45:27CDT2017
$ more plaintext5
plaintext5: No such file or directory
$ ./a.out
     +----------------------------------------------+
     |       Computer Science and Engineering       |
     |        CSCE 1030 - Computer Science I        |
     |   Student Name     EUID    euid@my.unt.edu   |
     +----------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? e
Enter the name of your input file you want to encrypt: plaintext5
```

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

```
ERROR: Unable to open file: plaintext5. Terminating...
$ more ciphertext5
ciphertext5: No such file or directory
$ ./a.out
     +---------------------------------------------+
     |       Computer Science and Engineering      |
     |        CSCE 1030 - Computer Science I       |
     |   Student Name     EUID    euid@my.unt.edu  |
     +---------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? d
Enter the name of your input file you want to decrypt: ciphertext5
ERROR: Unable to open file: ciphertext5. Terminating...
$ more key5
key5: No such file or directory
$ ./a.out
     +---------------------------------------------+
     |       Computer Science and Engineering      |
     |        CSCE 1030 - Computer Science I       |
     |   Student Name     EUID    euid@my.unt.edu  |
     +---------------------------------------------+

Would you like to ENCRYPT or DECRYPT a file (E or D)? d
Enter the name of your input file you want to decrypt: ciphertext1
Enter the name of the output file to write the plaintext: ptext1
Enter the file name that contains your encryption keys: key5
ERROR: Unable to open file: key5. Terminating...
```

TESTING:

Test your program to check that it operates as desired with a variety of inputs, especially boundary values or error conditions. Then, compare the answers your code gives with the ones you get from hand calculations.

Notice how, when using the same key file to encrypt and decrypt, an encrypted file will decrypt back to the original file. If you decrypt with a different key file than what was used to encrypt, you will get a different result.

If you want to use any of the plaintext or ciphertext files above for your testing, simply copy and paste the text into an editor on a CSE machine.

SUBMISSION:

Your program will be graded based largely upon whether it works correctly on the CSE machines, so you should make sure your program compiles and runs on the CSE machines.

Your program will also be graded based upon your program style. This means that you should use comments (as directed), meaningful variable names, and a consistent indentation style as recommended in the textbook and in class.

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.

CSCE 1030: Homework 4

- Program Header Example:

```
/*
 ============================================================================
 Name        : homework2.cpp
 Author      : Mark A. Thompson
 Version     :
 Copyright   : 2015
 Description : The program performs simple arithmetic operations based on in-
               put from the user.
 ============================================================================
 */
```

- Function Header Example:

```
/*
 ============================================================================
 Function    : deposit
 Parameters  : a double representing account balance and a double represent-
               ing the deposit amount
 Return      : a double representing account balance after the deposit
 Description : This function computes the account balance after a deposit.
 ============================================================================
 */
```

We will be using an electronic homework submission on Blackboard to make sure that all students hand their programming projects on time. You will submit both (1) the program source code file and (2) the algorithm design document to the **Homework 4** dropbox on Blackboard by the due date and time.

Note that this project must be done individually. Program submissions will be checked using a code plagiarism tool against other solutions, so please ensure that all work submitted is your own.

Note that the dates on your electronic submission will be used to verify that you met the due date and time above. All homework up to 24 hours late will receive a 50% grade penalty. Later submissions will receive zero credit, so hand in your best effort on the due date.

As a safety precaution, do not edit your program (using `vi` or `pico`) after you have submitted your program where you might accidentally re-save the program, causing the timestamp on your file to be later than the due date. If you want to look (or work on it) after submitting, make a copy of your submission and work off of that copy. Should there be any issues with your submission, this timestamp on your code on the CSE machines will be used to validate when the program was completed.

Copyright © 2017 by Mark A. Thompson. All rights reserved. No part of this document may be reproduced or distributed by any means in whole or in part without the express written consent of the author.