Steven Swanson

CSC 242 - Signature Assignment 3

Program: p8.7.cpp

Program will accept a key and input output files, an a flag to decode. If flag is not present, it will encrypt input file placing contents into output file. If decrypt flag is present, it will take encrypted input file and place decrypted contents into output file.

1. Prepare following variables
   1. Constant alphabet string - WIll be used for encryption/decryption and for preparing cipher.
   2. string for storing cipher named code
   3. boolean flag, to indicate if the program should be decrypting, start value False
   4. input file stream
   5. output file stream
2. process input arguments, for each argument (starting with index 1)
   1. if arg is equal to “-d” then set decrypt to True
   2. if arg starts with “-k” then using the remaining arg string, process the Cipher:
      1. create code string with value seed + alphabet in reverse
      2. loop over each character in the string, starting at index i=0
         1. loop over each character in the string, starting at j=i+1
         2. for each character, if the character is found twice, erase the character at position j.
      3. return the Cipher ( it will be code + reverse alphabet with duplicate character entries removed)
   3. if input file is not open, then argument must be input file
      1. open the file. If there is an error, write error and exit with return code 1
   4. if input file is open and output is not, then argument must be output file
      1. open the file. If there is an error, write error and exit with return code 1
3. Check to see that code contains a cipher and that input and output files are open.
   1. If any one is not present, write usage code, and exit with return code 1.
4. If decrypting, execute process\_file with input and output files, using code as input key and alphabet as output key which will output clear text.
5. if encrypting, execute process\_file with input and output files, using alphabet as input key and code as output key, which will output encrypted text.

process\_file :

Takes 4 inputs:

* Input file
* output file
* input\_key
* output\_key

process:

1. for each character read from input file:
2. if character is not alpha (A-Z,a-z) then write the character to output file.
3. if character is alpha, loop over input\_key until the character is found (all lower case)
4. using the index from step 3, write the character in output\_key at index. if character was uppercase, make output character uppercase as well. break the loop to start processing next character.