Assignment 10
Affine Cipher With Files
(Reading and Writing Files)

Modify Assignment 8, quiz3 (AKA examples 43 and 44), and example 46 so that the affine cipher is implemented with files. Your program will:

1. Use strings instead of c-strings.
2. Ask for an input file name and read it in using getline.
3. Ask for an output file name and read it in using getline.
4. Ask whether the user wants to encrypt or decrypt ('e', 'd') and read the answer using getline, repeatedly checking to be sure that the user has entered the proper values
5. Ask for and except an alpha value (the multiplier) where alpha is one of the positive odd integers < 26 except 13.
6. Ask for an except a beta value (the shift), where 0 <= beta <= 25
7. Write an encrypted (or decrypted) version of each alphabetic character found in the input file to the output file.
8. Using ex38.cpp as a model, check the input repeatedly so that alpha and beta are integers and in the proper range.
9. Load an array with all valid multiplicative inverses % 26 in order. That is, the multiplicative inverse of 1 is stored in the 0^{th} position, the multiplicative inverse of 3 in the 2^{nd} position, and so on.
10. Handle the issue of having a negative number % 26 with the function makePositive, described below

Extra Credit (5 points): your cipher text is a block of text 50 characters wide, with all spaces and punctuation removed, and with all alphabetic characters upper case.

Do this program one step at a time:

1. Write/compile/test the code necessary to process an input file.
2. Write/compile/test the code necessary to process an ouput file.
3. Write/compile/test the code necessary to input and check alpha and beta
4. Write/compile/test the code necessary to check whether 'e' or 'd' has been entered
5. Write/compile/check the function, makePositive
6. Insert the code from assignment 8. Only minimal modifications will be necessary.

If you don't follow these simple steps, you'll find yourself trapped in coding hell ("Abandon hope, all ye who enter here").

Pre: alpha is an integer % 26 with a multiplicative inverse
Post: returns the first positive alpha found by repeatedly adding 26 to alpha
int makePositive(int alpha);

Here are the functions from assignment 8 that are necessary.

Pre: inv is an integer array stored with multiplicative inverses % 26 in order. alpha is an
Integer % 26 with a multiplicative inverse
Post: returns the multiplicative inverse of alpha.
int multInv(int inv[], int alpha);

Pre: *ch* is an ascii-encoded character, *alpha* is the multiplier, *beta* is the shift amount
Post: if ch is alphabetic, returns the encrypted/decrypted ch using the affine cipher, else returns ch
Note: this function is the same for both encrypting and decrypting. The only difference is with the parameters. For decrypting: 1) alpha is the multiplicative inverse of the entered alpha; 2) beta is the negation of the multiplicative inverse of the entered alpha times the entered beta
char encDec(char ch, int alpha, int beta)

Pre: *ch* is an alphabetic character, *upper* is true if *ch* is uppercase, false otherwise
Post: The positional value of *ch* in the alphabet is returned. The positional value of 'A' and 'a' is 0.
int alphToInt(char ch, bool upper)

Pre: *pos* is in an integer in the range 0 <= *pos* <= 25, *upper* is true if the character associated with *pos* is upper case, false otherwise
Post: The alphabetic character associated with the positional value *pos* is returned.
char intToAlph(int pos, bool upper