Homework 1 - Permutation Cipher in C++

The permutation ciphers are used to generate a ciphertext (encrypted text) by re-arranging the position of the letters in a given plaintext.

How does it work?

1. Select the block size (n) and a permutation.
2. Divide a given plaintext into blocks of n letters and use 'x' as padding, if necessary.
3. Apply a permutation to each block and generate the encrypted ciphertext.

   For example, for n = 4 and permutation of (2, 1, 4, 3)
   Unencrypted plaintext is **gentlemen do not read each other's mail** [1]

```
1234 1234 1234 1234 1234 1234 1234 1234 1234 1234
gent leme n do  not  rea d ea ch o ther 's m ailx

2143 2143 2143 2143 2143 2143 2143 2143 2143 2143
egtn elem  nod n to r ae  dae hco  htre s'm  iaxl
```

   Encrypted ciphertext is **egtnelem nodn tor ae daehco htres'm iaxl**

The program needs to get the following information from the user:

- To encrypt: a text file that includes the plaintext, and a block size and the permutation information via standard input (keyboard).
- To decrypt: a binary file that includes an encrypted ciphertext, and a block size and the permutation information via standard input (keyboard).

Here's what a sample run should look like:

```
> ./PermCipher
Usage: ./PermCipher option -i InputFileName -o OutputFileName
Options: -e Encrypt
         -d Decrypt


> ./PermCipher -e -i PlaintextFile -o CiphertextFile
Welcome to the Permutation Cipher
Selected Mode: Encrypt
Input File: PlaintextFile
Output File: CiphertextFile
Please enter the block size (2-8) and the permutation (e.g., 4 2413): 4 2143
Encrypted ciphertext file: egtnelem nodn tor ae daehco htres'm iaxl

> ./PermCipher -d -i CiphertextFile -o PlaintextFile
Welcome to the Permutation Cipher
Selected Mode: Decrypt
Input File: CiphertextFile
Output File: PlaintextFile
Please enter the block size (2-8) and the permutation (e.g., 4 2413): 4 2143
Decrypted ciphertext file: gentlemen do not read each other's mail
```

The program should be able to process up to 8 characters at each block with a minimum of 2 characters. Submit your makefile, source code (encrypt.h encrypt.cpp, decrypt.h, decrypt.cpp, main.cpp) and test files to your BitBucket repository csci221-HW01.

[1] https://en.wikipedia.org/wiki/Black_Chamber