# CST8234 – C Programming
## Assignment 03B: Encryption Algorithms

## Transposition Algorithm

In this a *transposition* encryption algorithm, each character's position is shifted to a new location within the file.

Imagine that the encryption key is 3. The conversion process starts by outputting the first character of the *plain text* file to the *cipher text* file. Next, it offsets the position in the *plain text* file by 3 (the encryption key value), that is the 4th position, and outputs the 4th *plain text* character to the *cipher text* file as the second *cipher text* character.

Again, it offsets by another 3 positions, that is, it accesses the 7th letter in the *plain text* file, and outputs it as a third character in the *cipher text* file.

The process continues until the algorithm reaches the end of the *plain text* file. But this has output only a small subset of the original *plain text* file (every third character given a key of 3). The next pass through the *plain text* file starts at character 2, then character 5, then character 8 (and so on). The final pass through the file starts at character 3, then 6, then 9 (and so on).

Whatever value you use as the key will determine the number of passes that must be made through the original *plain text* file.

The following *plain text*:

```
Testing1234567890ABCDEF
```

would result in the following *cipher text* given a key value of 3 (the conversion mapping is shown in the diagram below):

```
Ttg369BEei1470CFsn258AD
```

Given a key value of 11, the cipher text would be:

```
T5Fe6s7t8i9n0gA1B2C3D4E
```

You are to write a small C program `transpo`, to encrypt / decrypt a file using the above algorithm. The program should have the following functionality:

```
transpo [ OPTIONS ] SOURCE DESTINATION
```

**`OPTIONS:`**

`-d` KEY

decrypt the file SOURCE using KEY and writes back into DESTINATION

`-e` KEY

encrypt the file SOURCE using KEY and writes back into DESTINATION

`-h`: help in using the command

Option `d` and `e` are exclusive, if `d` is on, `e` can not be on, and both of them require an argument, the encryption KEY

In case that the user does not give all the required command arguments, you program should print a usage message and exit with EXIT_FAILURE.

## Assignment Progression

Now that you have finished your first algorithm, use your **`transpo`** and the KEY **`8234`**, to **`decrypt`** the file **`mystery02`** file that you'll find in BB.