CS19001: Programming and Data Structures Laboratory

Assignment No. 9 (Handling File and Command-Line Arguments)

Date: 18-October-2019

Problem Statement:

In the city of Wonderland, Alice used to send one leaflet of information to her elder sister Lory on every evening through the hands of the Dormouse. However, they do not want that the Dormouse can read out the information that they shared. So, Alice uses a simple technique of encrypting the plain text by replacing every character in the plain text by a character which is fixed number of positions down the list of the alphabet. The last characters are folded back to the beginning. She uses a random number (between $1 - 10$) for this shift and tell this number to Lory beforehand. The original text, from where Alice encrypts, is wrote with only alphabetic, numeric and symbolic characters. Finally, Dory sees the encrypt message and understands it fully as she knew the protocol (shift). Then, she also notes all the different alphabets, numbers and symbols (except spaces) used in the encrypted message.

Your task is to write a C-program using command-line arguments to simulate this whole process.

- You have to read the text information from an existing input file which indicates the original message that needs to be communicated.
- You have to convert the text information from the input file into encrypted message and write it into another intermediate file, named "inter.txt" (let the file-name be fixed). The shift to be given (that generates the encryption) is a random number (within $1 - 10$) and only applied to alphabets.
- Finally, you have to produce another new output file containing a note of all the different alphabets, numbers and symbols (write '`\n`' for newlines and '`\t`' for tabs encountered and ignore the spaces) used in the encrypted message, only if (and when) the command line argument $-$`flag` indicates `a`, `n`, `s`, respectively. All these options are optional here, so as the $-$`flag` entry as well. For example, if only $-$`flag n` is specified, then you should only write the numeric characters into the output file.

To generate the random number between $0 - 5$, please go through the following code snippet:

```
#include <stdio.h>
#include <stdlib.h>    // for srand(), rand()
#include <sys/types.h> // for getpid()
#include <unistd.h>    // for getpid()

int main()
{
    // declare srand() in the beginning of main (only once)
    srand( getpid() );
    // generate any random number [0-5]
    x = rand() % 5;
    return 0;
}
```

You need to use this to generate your random shift required. Moreover, the command-line arguments that you may use is as follows ([ ] indicates optional arguments):

```
./a.out -input <Input_File_Name> [-output <Output_File_Name>] [-flag <Flag_Option>]
```

Note that, the order of giving the arguments may be different, but the `Input_File_Name`, `Output_File_Name` and `Flag_Option` must succeed $-$`input`, $-$`output` and $-$`flag` arguments, respectively. Also, it is left to you to check for correctness in command-line arguments and report any warnings/errors (check Samples 2-5).

Example Inputs/Outputs:

Sample-1:
Sample-Command:

```
./a.out  -input  input.txt  -output  output.txt  -flag  ans
```

Sample-Input-File: (Filename: input.txt)

```
00 : A Quick Brown Fox, Jumps Over The Lazy Dog ...
01 : Alice's Adventures In Wonderland !
```

Sample-Intermediate-File: (Filename: inter.txt)

```
00 : D Txlfn Eurzq Ira, Mxpsv Ryhu Wkh Odcb Grj ...
01 : Dolfh'v Dgyhqwxuhv Lq Zrqghuodqg !
```

Sample-Output-File: (Filename: output.txt)

```
Alphabetic Characters:
D T x l f n E u r z q I r a M x p s v R y h u W k h O d c b G r j D o l f h v D
g y h q w x u h v L q Z r q g h u o d q g
Numeric Characters:
0 0 0 1
Symbolic Characters:
: , . . . \n : ' ! \n \n
```

Sample-2:
Sample-Command:

```
./a.out  -input  -output  output.txt  -flag  ans
```

Output:

```
Error: Input_File_Name Not Specified!!
USAGE:: ./a.out -input <Input_File_Name> [-output <Output_File_Name>] [-flag <Flag_Option>]
        ([ ] indicates optional arguments)
```

Sample-3:
Sample-Command:

```
./a.out  -in  input.txt  -out  output.txt  -flag  ans
```

Output:

```
Error: Wrong Arguments!!
USAGE:: ./a.out -input <Input_File_Name> [-output <Output_File_Name>] [-flag <Flag_Option>]
        ([ ] indicates optional arguments)
```

Sample-4:
Sample-Command:

```
./a.out  -input  input.txt  -output  output.txt  -flag  ans  123
```

Output:

```
Error: Too Many Arguments!!
USAGE:: ./a.out -input <Input_File_Name> [-output <Output_File_Name>] [-flag <Flag_Option>]
        ([ ] indicates optional arguments)
```

Sample-5:
Sample-Command:

```
./a.out  input.txt
```

Output:

```
Error: Too Few Arguments!!
USAGE:: ./a.out -input <Input_File_Name> [-output <Output_File_Name>] [-flag <Flag_Option>]
        ([ ] indicates optional arguments)
```

Submit a single C source file. Do not use global/static variables.