ECE 3331, Fall 2023 Programming Assignment 02, due Sunday 09/03 at 11:59 pm

Turn in this program to the blackboard course website by the due date.

Write a program to

(1) Open an existing input file called "in.txt" for reading, and open a new file called out.txt for writing.
The file "in.txt" will be located in the same directory from which the executable from your program will be run.
That means, in your fopen( ) ansi c function, you do not need to list the full pathname of the file to be opened, only the filename itself.
The file "in.txt" will contain a list of integers, one integer per line.
If the file "in.txt" does not open successfully when your program attempts to open it, print a message to the screen that the file "in.txt" does not exist, and exit the program.
One way to check if the file was successfully opened is the following.
If the statement fpi=fopen("ssn_in.txt","r") does not successfully open the file "in.txt" for reading,
then the FILE pointer fpi is equal to the macro NULL.
That is,
if(fpi = = NULL)printf("The file in.txt did not successfully open");

(2) Prompt the user of your program to enter an integer and read-in that integer from the keyboard.

(3) Copy the integers from the file “in.txt” to the file “out.txt”.
But do not copy any occurrences of the user-entered integer from “in.txt” to “out.txt”.
Additionally, count the number of occurrences of the user-entered integer in the file “in.txt”.

You might use a while( ) loop to read the integers from the file.
Recall that fscanf( ) returns a number when it executes. If fscanf( ) tries to read from a file, but everything in the file has already been read, then fscanf( ) returns the value EOF which is defined in stdio.h. On Windows machines, typically, EOF stands for -1. For xcode on Apple machines, EOF =0.
So for the logical expression in the while( ) loop, test whether the ireturn=fscanf( ) is not equal to EOF

```
ireturn=fscanf( …..);
while(ireturn != EOF){
   if( …){
     …..
     fprintf(… );
  }
  ireturn=fscanf( …..);
}
```

(4) Print a message to screen informing the user about the total number of occurrences of the user-entered integer in the file.

In order to test your program, a file “in.txt” containing some arbitrary integers is included in the Blackboard assignment panel.
Download the file and place it in the directory (folder) containing your executable *.exe program file.
Alternatively, you can use the Microsoft text editor notepad.exe to create a file called in.txt; then
write some integer values into the file and place the file in the folder containing your program executable file *.exe.
Another way is to simply write a program to create a file “in.txt” and to fprintf( ) some integers into the file, and close it.