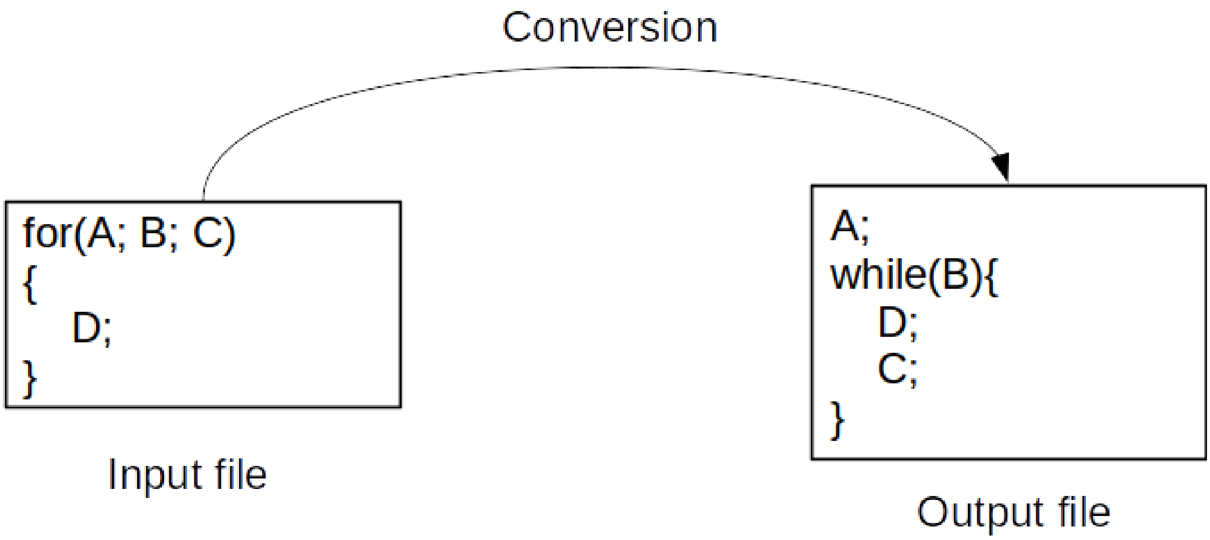
KTU BUL 112: Introduction to computer programming.
Assignment 3: "Loop Conversion"

Purpose
The purpose of this project is to get you familiar with file I/O and string manipulations.

Rules
Make sure you do coding of your program on your own. You can ask for help from your friends to understand the problem, figure out how to tackle the problem, but you may never copy someone's code and present it as your own. If this is detected, you will receive a caution, your code will not be marked. If you copied your work from a friend of yours, you will both receive a caution and your work will not be marked.

The Task

Your task is to write a C code which will read C statements representing a “for” loop from an input file, convert it to its “while” equivalent, and then store it back into an output file as shown in Fig. 1.



```
for(A; B; C)
{
    D;
}
```

```
A;
while(B){
    D;
    C;
}
```

Note that the opening and closing braces of the body of the loop in input file may appear on any location which is syntactically correct. Furthermore, there may be two nested “for“ loops in input file in which case your program is expected to convert both of them. Your program should terminate with an error message, if an incorrect input (e.g. a missing closing brace or semicolon) is provided. Your program should take the names of the input and output files as two different parameters from the command line:

```
$> ./convert input.txt output.txt
```

Step 5: Create a readme File

Use an editor to create a text file named readme (not readme.txt, or README, or Readme, etc.) that contains:

- Your name, student ID, and the assignment number.

- A description of whatever help (if any) you received from others while doing the assignment, and the names of any individuals with whom you collaborated, as prescribed by the course Policy web page.

- (Optionally) An indication of how much time you spent doing the assignment.

- (Optionally) Your assessment of the assignment: Did it help you to learn? What did it help you to learn? Do you have any suggestions for improvement? etc.

- (Optionally) Any information that will help us to grade your work in the most favorable light. In particular you should describe all known bugs.

Descriptions of your code should not be in the readme file. Instead **they should be integrated into your code as comments.**

Your readme file should be a plain text file. Don't create your readme file using Microsoft Word, Hangul (HWP) or any other word processor.

Step 6: Submit

Your submission should include your zipped codeblocks project files and your readme file. You can do that using Moodle platform.

Grading

We will grade your work on two kinds of quality: quality from *the user's* point of view, and quality from *the programmer's* point of view.
From the user's point of view, a program has quality if it behaves as it should.
From the programmer's point of view, a program has quality if it is well styled and thereby easy to maintain. For this assignment we will pay particular attention to above rules. These additional rules apply:

- **Names**: You should use a clear and consistent style for variable and function names. One example of such a style is to prefix each variable name with characters that indicate its type.

- **Comments**: Each source code file should begin with a comment that includes your name, the number of the assignment, and the name of the file.

- **Comments**: Each function -- especially the main function -- should begin with a comment that describes *what the function does* from the point of view of the caller. (The comment should not describe *how the function works*.) It should do so by *explicitly* referring to the function's parameters and return value. The comment also should state what, if anything, the function reads from the standard input stream or any other stream, and what, if anything, the function writes to the standard output stream, the standard error stream, or any other stream. Finally, the function's comment should state which global variables the function uses or affects. In short, a function's comments should describe the flow of data into and out of the function.

- **Line lengths**: Limit line lengths in your source code to 72 characters. Doing so allows us to print your work in two columns, thus saving paper.