COMP 451

FORMAN CHRISTIAN COLLEGE

(A CHARTERED UNIVERSITY)

COMPILER CONSTRUCTION

Programming Assignment 1

It's an open books and open notes assignment. Use of Internet is allowed. This assignment can be done in groups of MAX two students. You CANNOT share your code with any other group. Any such attempt will be dealt with seriously.

Grading Criteria

Working Code: 60%

Properly formatted Report: 20%

Viva: 20%

Important: You need to submit a well formatted and well written report for this assignment. The report should carry following sections:

- Introduction about the problem in hand especially well written information about the preprocessor and its functions.
- Detailed and easy to understand description of your logic. Make separate section for each component. In this assignment at least three sections describing each function.
- Start early. **NO additional time in any case what so ever will be granted. A penalty of 60% will be applied on late submission.**
- Viva will be conducted for this assignment. Date will be announced later.

Hard Deadline: Code file should be submitted on or before 8:00 am Mon May 23, 2022 on MOODLE course page. A hard copy of the report should be submitted on same day in class. (Without code and screen shot of output)

You need to submit a hard copy of your report on the same day (May 23, in class)

DONOT add code to the hardcopy of report that you will submit.

Your submission on MOODLE should carry

- **The code file.**
- **All the output files. (intermediate output files as well as the final output file)**
- **Your report. Report should be like documentation. Clearly describing your work. Marks for report are based on this.**
- **You will be called for a viva based on what you have submitted.**

Assignment Task [60 Marks]

In this task we will write a preprocessor. Your program should accept a C file from command line. The file

should contain a valid C program. You need to write a single C file for this assignment with minimum three functions:

- void stripOffComments(. . .)
- void macroExpansion(. . .)
- void include HeaderFiles(. . .)

Please make sure to use the same naming convention. You can use other user defined function within the functions given above as per your logic.

void stripOffComments(. . .)

In this program, you will write a program that reads a .c file provided by user on command line. It then calls two functions one after the other. These functions get rid of double slash and slash-star comments from the input file respectively. Your program should also remove extra blank lines (if any) in the program. The output of the program should be another C file names out1.c.

void macroExpansion(. . .)

This function takes a file , out1.c (final output of stripOffComments.c program) as input. You may pass this as argument to your function.

Program then searches the file for macro line/s. Note that we may have one or more macro lines in our program. For example in our sample input program there is a macro line that reads:

#define MSG “Invalid Arguments\n”

Once you have that line with you, your program should then parse the string and identify the head and body of macro. Here we assume that head of macro is `MSG` and body is `“Invalid Arguments\n”.`

Your program should then search for the string in the input file that should match head of the macro and replace it with the macro body.

Finally your program should write the output in the another file, out2.c.

Make sure that the macro line/s should not be copied in the output file.

void includeHeaderFiles(…)

This function will accept the output file out2.c as input. You may pass the file as function argument.

Your function will scan the input file and look for the header files that are included in the input program. Note that out2.c is the input file to this function.

There are three header files accompanying this assignment which are scrapped versions of stdio.h, stdlib.h and string.h. Your program should identify which of these header file is included in the input file and should replace the `#include <. . .>` statement with the corresponding file/s, and write the new contents in another file. This is your final output file. Suppose the name of this file is final.c

Your program should finally display the contents of final.c on console.

Few input file samples are shown for your convenience.

Input file 1

```
/*****in1.c*****/
#include <stdio.h>
//defining macros
#define ON  1
#define OFF  0

void main()
{

      /*declaring variables*/
      int j = 2;
      int motor,sensorValue = 0;
      if(motor == ON)
      {
            sensorValue++;
      }
      else if(motor == OFF)
      {
            sensorVlaue--;
      }
      return 0;
}
```

Input file 2

```
/*Program Name: in1.c
```

```
*Description:
      Input file for assignment 1
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
//defining macros
#define MESSAGE1       “Hello class\n”
#define Message2       “Computer Science Department”

void main()
{

      ////////////Print messages

      printf(Message1);
      printf(Message2);



      return 0;
}
```

Input file 3

```
/*Program Name: in1.c
*Description:
      Input file for assignment 1
*******************************/
#include <stdio.h>
#include <string.h>
#define     TRUE  1
/*this file carries two MACROS*/
int main(int argc,char*argv[])
{

      //checking the arguments
      if(argc!=2)
      {
            printf("Error encountered\n");
            exit(0);//terminate the program
      }
      int i = strlen(argv[1];
      if(i >= 10)
      {
            //do something
      }
      While(TRUE)
      {
            /*this is an infinite loop and will
            Iterate till the user terminates the program
            Using Ctrl+C*/
      }
      //end of program


      return 0;
}
```

Input file 4

```
//in1.c
```

```
//input file for assignment 1
#include <stdio.h>

#include <stdlib.h>


#define     CODE  “COMP 451\n”

int main(int argc,char*argv[])
{

      //checking the arguments
      if(argc!=2)
      {
            printf(“Error encountered\n”);
            exit(0);//terminate the program
      }
      printf(“Course Code provided by user is: “);
      printf(CODE);
      int j = strlen(argv[1];
      int i = 0;
      While(i<j)
      {
            /*this is a finite loop and will
            iterate till the value of i remains less
            than j*/

            i++;//must do this else it iterates forever
      }
      //end of program


      return 0;
}
```