COMPILER DESIGN LAB-2 SUBMISSION

NAME : DEVADATHAN N R

ROLL NO : 04

SECTION : CSE A1

REG NO : 230905010

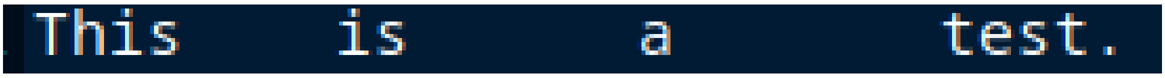
WEEK 2

Question 1:
That takes a file as input and replaces blank spaces and tabs by single space and writes the output to a file.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    FILE *inputFile, *outputFile;
    int ch;
    int prevSpace = 0;

    if (argc != 3)
    {
        printf("Usage: %s <input_file> <output_file>\n", argv[0]);
        return 1;
    }

    inputFile = fopen(argv[1], "r");
    if (inputFile == NULL)
    {
        printf("Error: Could not open input file.\n");
        return 1;
    }

    outputFile = fopen(argv[2], "w");
    if (outputFile == NULL)
    {
        printf("Error: Could not open output file.\n");
        fclose(inputFile);
        return 1;
    }
    while ((ch = fgetc(inputFile)) != EOF)
    {
        if (ch == ' ' || ch == '\t')
        {
            if (!prevSpace)
            {
                fputc(' ', outputFile);
                prevSpace = 1;
            }
        }
        else
        {
            fputc(ch, outputFile);
            prevSpace = 0;
        }
    }

    fclose(inputFile);
    fclose(outputFile);

    printf("File processed successfully.\n");
    return 0;
}
```

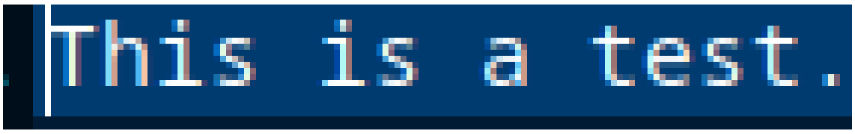
Output:

input.txt



```
This    is       a        test.
```

output.txt



```
This is a test.
```

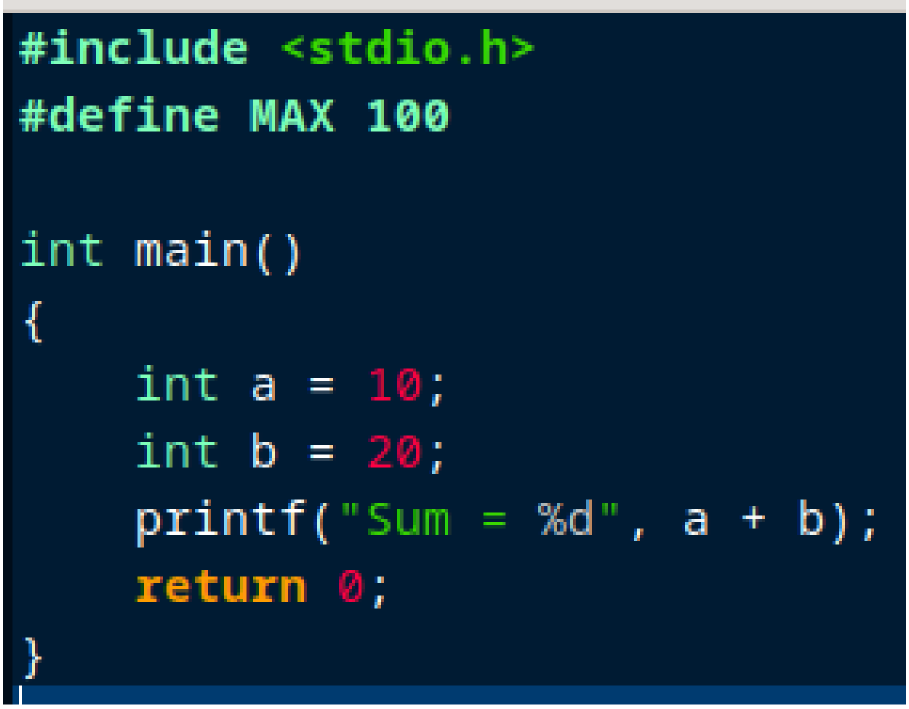
Question 2:

To discard preprocessor directives from the given input ‘C’ file.

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    FILE *inputFile, *outputFile;
    char line[1000];

    if (argc != 3)
    {
        printf("Usage: %s <input_file.c> <output_file.c>\n", argv[0]);
        return 1;
    }

    inputFile = fopen(argv[1], "r");
    if (inputFile == NULL)
    {
        printf("Error: Cannot open input file.\n");
        return 1;
    }

    outputFile = fopen(argv[2], "w");
    if (outputFile == NULL)
    {
        printf("Error: Cannot open output file.\n");
        fclose(inputFile);
        return 1;
    }

    while (fgets(line, sizeof(line), inputFile))
    {
        if (line[0] != '#')
        {
            fputs(line, outputFile);
        }
    }

    fclose(inputFile);
    fclose(outputFile);

    printf("Output File Ready! \n");
    return 0;
}
```

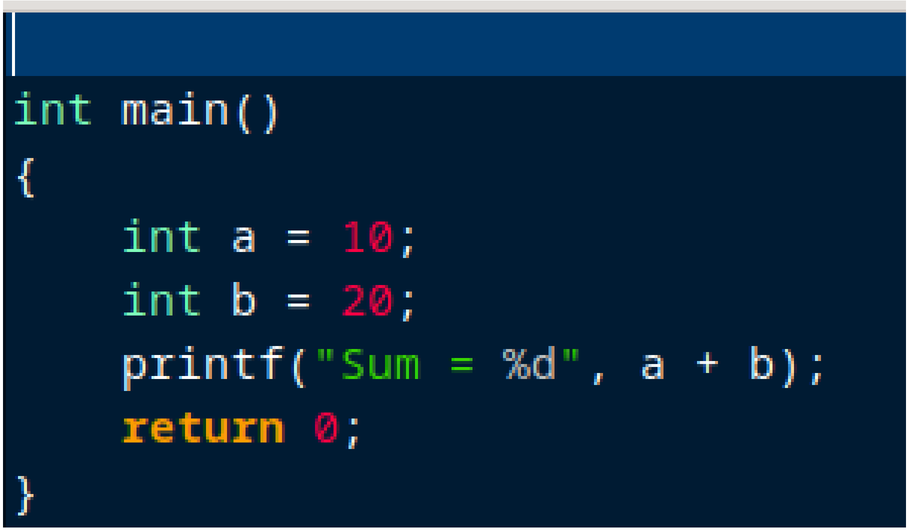
Output:

input.c



```
#include <stdio.h>
#define MAX 100

int main()
{
    int a = 10;
    int b = 20;
    printf("Sum = %d", a + b);
    return 0;
}
```

output.c



```

int main()
{
    int a = 10;
    int b = 20;
    printf("Sum = %d", a + b);
    return 0;
}
```

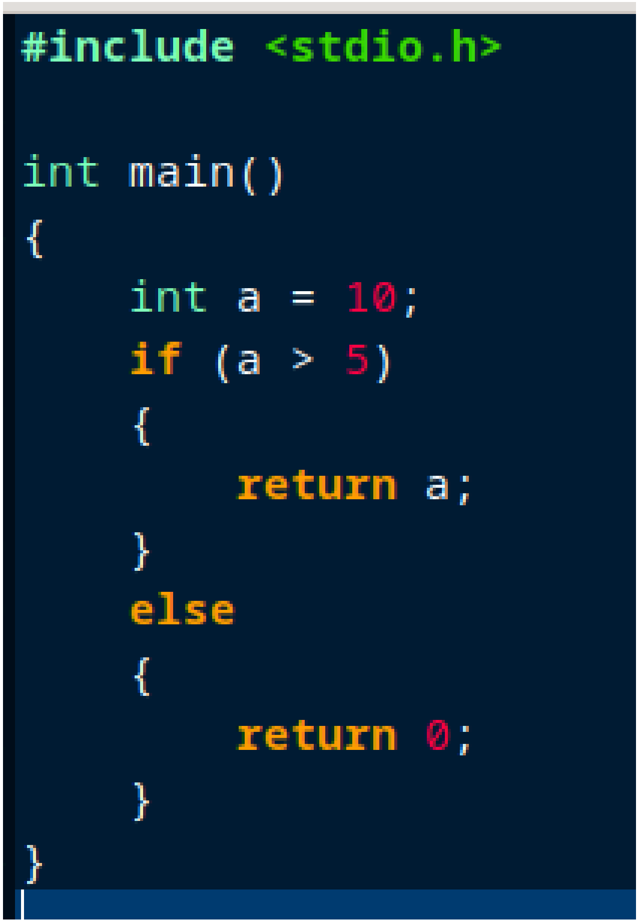
All lines starting with ‘#’ has been removed, after reading line by line.

Question 3:

That takes C program as input, recognizes all the keywords and prints them in upper case.

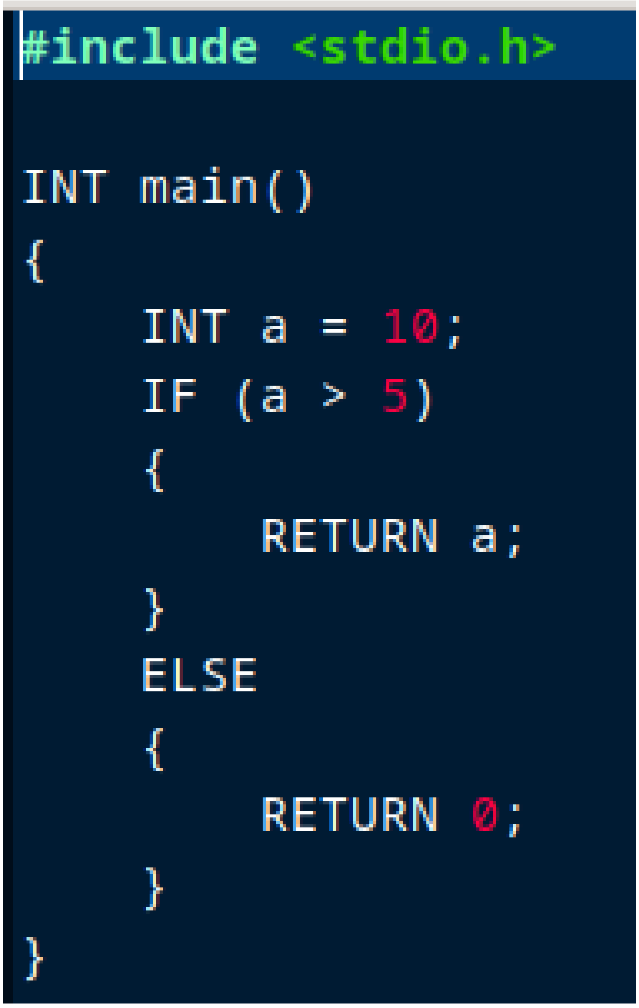
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

const char *keywords[] = {
    "char", "do", "double", "else”, "float",
    "for","if", "int", "long", "return"
};

#define KEYWORD_COUNT 10

int isKeyword(char *word)
{
    for (int i = 0; i < KEYWORD_COUNT; i++)
    {
        if (strcmp(word, keywords[i]) == 0)
            return 1;
    }
    return 0;
}

void toUpperCase(char *word)
{
    for (int i = 0; word[i]; i++)
        word[i] = toupper(word[i]);
}

int main(int argc, char *argv[])
{
    FILE *inputFile, *outputFile;
    char ch, word[100];
    int index = 0;

    if (argc != 3)
    {
        printf("Usage: %s <input.c> <output.c>\n", argv[0]);
        return 1;
    }
    inputFile = fopen(argv[1], "r");
    outputFile = fopen(argv[2], "w");

    if (inputFile == NULL || outputFile == NULL)
    {
        printf("Error opening file.\n");
        return 1;
    }

    while ((ch = fgetc(inputFile)) != EOF)
    {
        if (isalnum(ch) || ch == '_')
        {
            word[index++] = ch;
        }
        else
        {
            if (index > 0)
            {
                word[index] = '\0';

                if (isKeyword(word))
                { toUpperCase(word); }

                fputs(word, outputFile);
                index = 0;
            }

            fputc(ch, outputFile);
      }  }

    if (index > 0)
    {
        word[index] = '\0';
        if (isKeyword(word))
        {
            toUpperCase(word);
        }
        fputs(word, outputFile);
    }

    fclose(inputFile);
    fclose(outputFile);
    return 0;
}
```

input.c



```c
#include <stdio.h>

int main()
{
    int a = 10;
    if (a > 5)
    {
        return a;
    }
    else
    {
        return 0;
    }
}
```

output.c



```c
#include <stdio.h>

INT main()
{
    INT a = 10;
    IF (a > 5)
    {
        RETURN a;
    }
    ELSE
    {
        RETURN 0;
    }
}
```