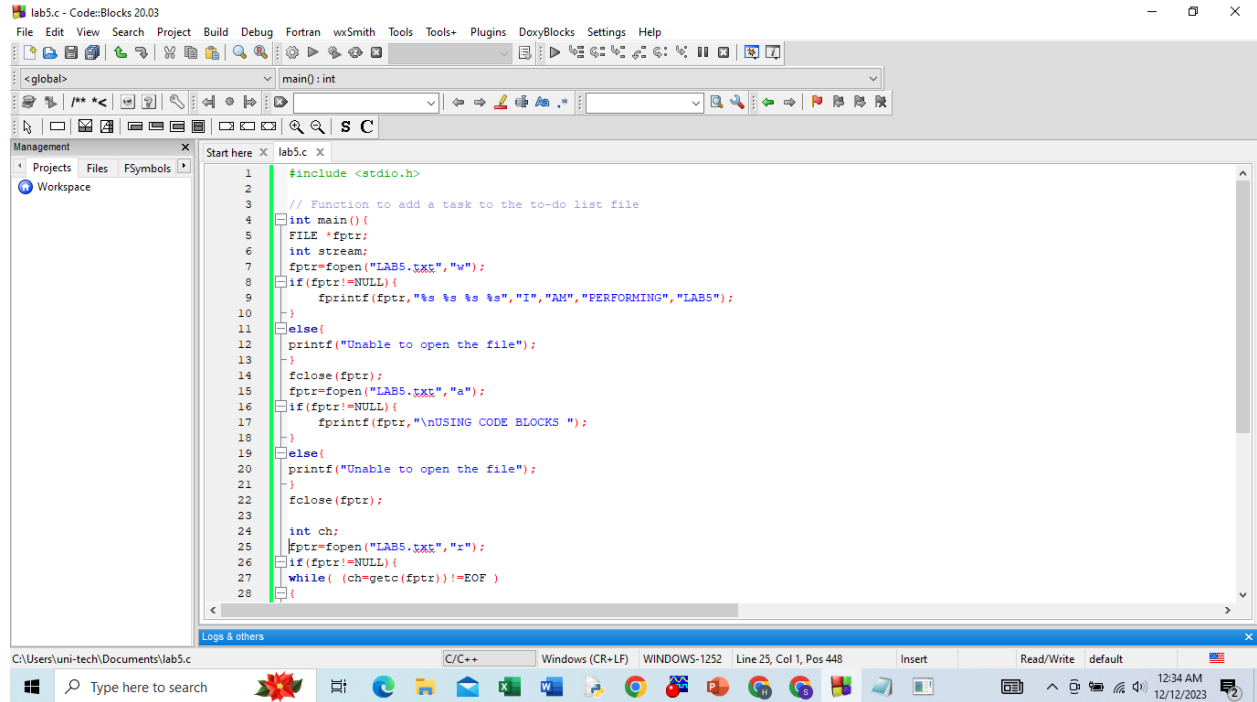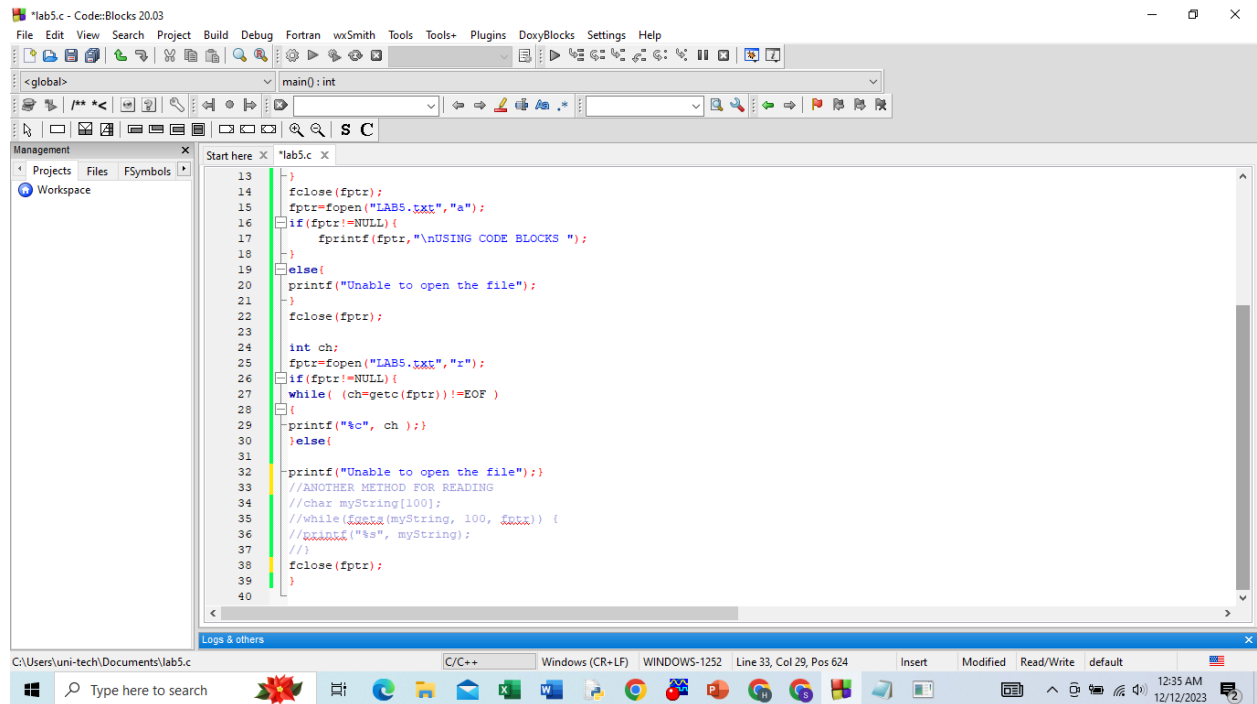# LAB 5

Q1:
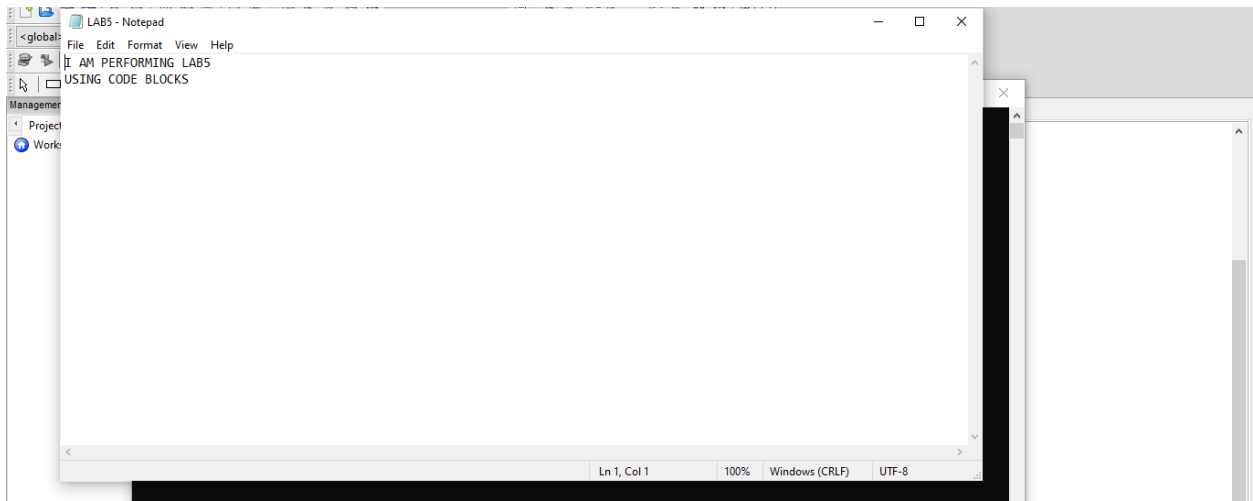


```c
#include <stdio.h>

// Function to add a task to the to-do list file
int main(){
    FILE *fptr;
    int stream;
    fptr=fopen("LAB5.txt","w");
    if(fptr!=NULL){
        fprintf(fptr,"%s %s %s %s","I","AM","PERFORMING","LAB5");
    }
    else{
        printf("Unable to open the file");
    }
    fclose(fptr);
    fptr=fopen("LAB5.txt","a");
    if(fptr!=NULL){
        fprintf(fptr,"\nUSING CODE BLOCKS ");
    }
    else{
        printf("Unable to open the file");
    }
    fclose(fptr);

    int ch;
    fptr=fopen("LAB5.txt","r");
    if(fptr!=NULL){
    while( (ch=getc(fptr))!=EOF )
    {
```
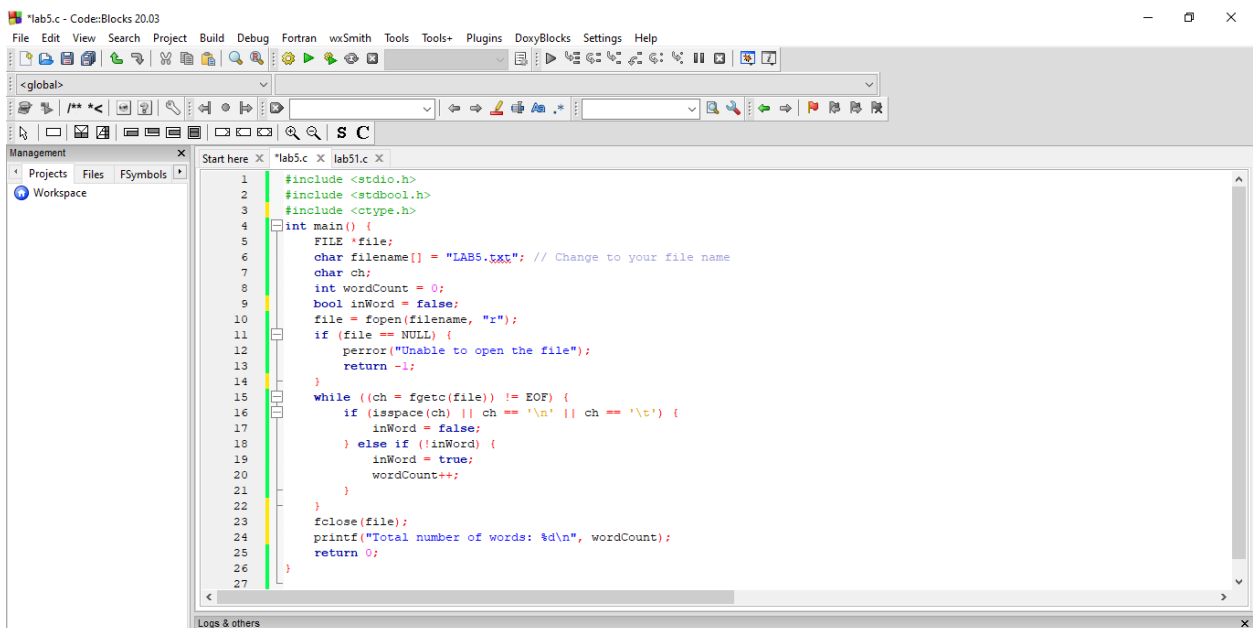
```c
    }
    fclose(fptr);
    fptr=fopen("LAB5.txt","a");
    if(fptr!=NULL){
        fprintf(fptr,"\nUSING CODE BLOCKS ");
    }
    else{
        printf("Unable to open the file");
    }
    fclose(fptr);

    int ch;
    fptr=fopen("LAB5.txt","r");
    if(fptr!=NULL){
    while( (ch=getc(fptr))!=EOF )
    {
    printf("%c", ch );}
    }else{

    printf("Unable to open the file");}
    //ANOTHER METHOD FOR READING
    //char myString[100];
    //while(fgets(myString, 100, fptr)) {
    //printf("%s", myString);
    //}
    fclose(fptr);
    }
```

OUTPUT:
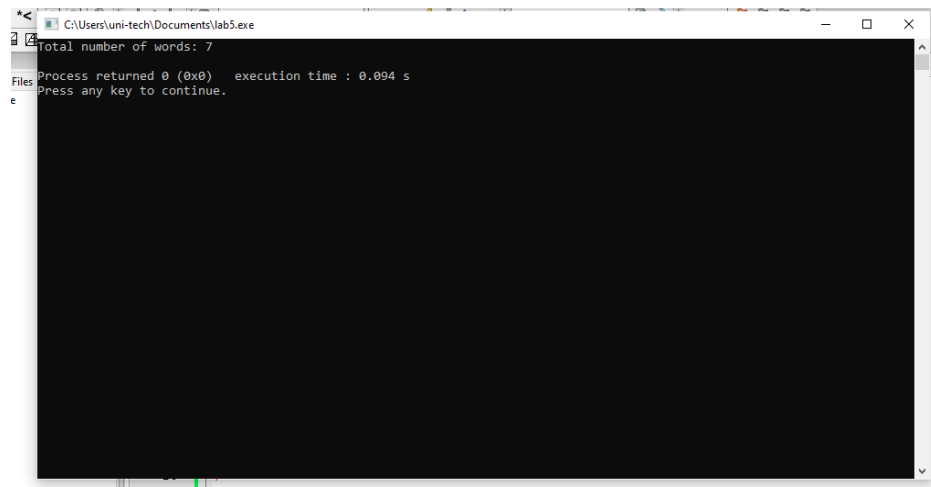
LAB5 - Notepad

File   Edit   Format   View   Help

I AM PERFORMING LAB5
USING CODE BLOCKS
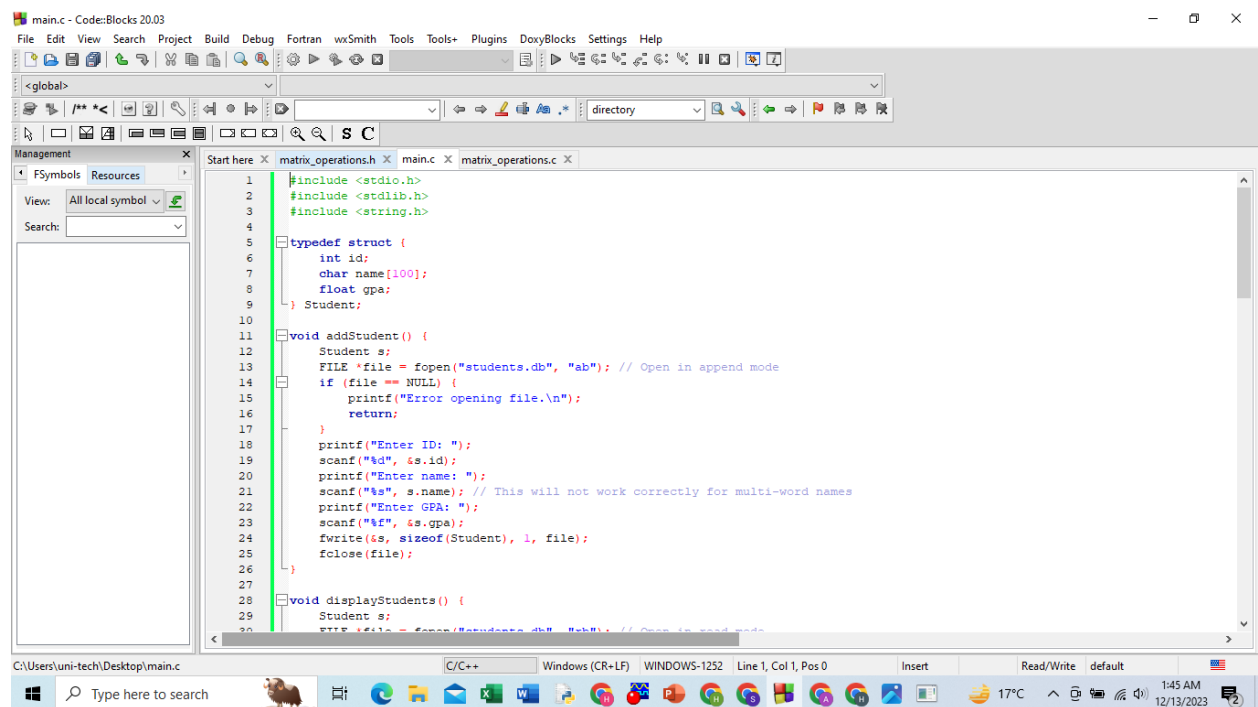
Ln 1, Col 1   100%   Windows (CRLF)   UTF-8

Q2:

*lab5.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Start here   *lab5.c   lab51.c

```c
1    #include <stdio.h>
2    #include <stdbool.h>
3    #include <ctype.h>
4    int main() {
5        FILE *file;
6        char filename[] = "LAB5.txt"; // Change to your file name
7        char ch;
8        int wordCount = 0;
9        bool inWord = false;
10       file = fopen(filename, "r");
11       if (file == NULL) {
12           perror("Unable to open the file");
13           return -1;
14       }
15       while ((ch = fgetc(file)) != EOF) {
16           if (isspace(ch) || ch == '\n' || ch == '\t') {
17               inWord = false;
18           } else if (!inWord) {
19               inWord = true;
20               wordCount++;
21           }
22       }
23       fclose(file);
24       printf("Total number of words: %d\n", wordCount);
25       return 0;
26   }
27
```

Logs & others

OUTPUT:

Q3:


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

typedef struct {
    int id;
    char name[100];
    float gpa;
} Student;

void addStudent() {
    Student s;
    FILE *file = fopen("students.db", "ab"); // Open in append mode
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }
    printf("Enter ID: ");
    scanf("%d", &s.id);
    printf("Enter name: ");
    scanf("%s", s.name); // This will not work correctly for multi-word names
    printf("Enter GPA: ");
    scanf("%f", &s.gpa);
    fwrite(&s, sizeof(Student), 1, file);
    fclose(file);
}

void displayStudents() {
    Student s;
    FILE *file = fopen("students.db", "rb"); // Open in read mode
```

Top window — main.c (lines 29–57):

```c
        Student s;
        FILE *file = fopen("students.db", "rb"); // Open in read mode
        if (file == NULL) {
            printf("Error opening file.\n");
            return;
        }
        while (fread(&s, sizeof(Student), 1, file)) {
            printf("ID: %d, Name: %s, GPA: %.2f\n", s.id, s.name, s.gpa);
        }
        fclose(file);
    }

void deleteStudent(int id) {
        Student s;
        FILE *file = fopen("students.db", "rb"); // Open in read mode
        FILE *temp = fopen("temp.db", "wb"); // Temporary file
        if (file == NULL || temp == NULL) {
            printf("Error opening file.\n");
            return;
        }
        while (fread(&s, sizeof(Student), 1, file)) {
            if (s.id != id) {
                fwrite(&s, sizeof(Student), 1, temp);
            }
        }
        fclose(file);
        fclose(temp);
        remove("students.db");
        rename("temp.db", "students.db");
```

Bottom window — main.c (lines 54–82):

```c
        fclose(file);
        fclose(temp);
        remove("students.db");
        rename("temp.db", "students.db");
    }

void modifyStudent(int id) {
        Student s;
        FILE *file = fopen("students.db", "rb+");
        if (file == NULL) {
            printf("Error opening file.\n");
            return;
        }
        while (fread(&s, sizeof(Student), 1, file)) {
            if (s.id == id) {
                printf("Enter new name: ");
                scanf("%s", s.name); // This will not work correctly for multi-word names
                printf("Enter new GPA: ");
                scanf("%f", &s.gpa);
                fseek(file, -sizeof(Student), SEEK_CUR); // Move back to the beginning of the current record
                fwrite(&s, sizeof(Student), 1, file);
                break;
            }
        }
        fclose(file);
    }

int main() {
    int choice, id;
    while (1) {
```

```c
        int choice, id;
        while (1) {
            printf("\n1. Add Student\n2. Display Students\n3. Delete Student\n4. Modify Student\n5. Exit\nEnter choice: ");
            scanf("%d", &choice);
            switch (choice) {
                case 1:
                    addStudent();
                    break;
                case 2:
                    displayStudents();
                    break;
                case 3:
                    printf("Enter ID to delete: ");
                    scanf("%d", &id);
                    deleteStudent(id);
                    break;
                case 4:
                    printf("Enter ID to modify: ");
                    scanf("%d", &id);
                    modifyStudent(id);
                    break;
                case 5:
                    exit(0);
                default:
                    printf("Invalid choice\n");
            }
        }
    }
    return 0;
}
```

OUTPUT: