



PROGRAMMING FUNDAMENTALS

- LAB

Name: Mohsin Ali
Roll Number: 25P-0545

Q#1: Write a C program to read name and marks of n number of students and store them in a file

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows a folder structure under LAB containing .vscode, task1, task1.c, task1.exe, task1.txt, task2, task2.c, task2.txt, task3, task3.bin, task3.c, task4, task4_1.txt, task4_2.txt, task4_1.txt, task4_2.txt, task5, task5.c, task5.txt, task6, task6.c, and task6.txt.
- CODE EDITOR:** Task1.c is open, displaying the following C code:

```
//Write a C program to read name and marks of n number of students and store them in a file
//Name: Mohsin Ali
//Roll Number: 25P-0545

#include <stdio.h>
#include <string.h>

int main()
{
    printf("Name: Mohsin Ali\nRoll Number: 25P-0545\n");

    int n;
    printf("Enter number of students: ");
    scanf("%d",&n);
    // int name[n][256];

    getchar();
    FILE *ptr=fopen("task1.txt","w");
}
```
- TERMINAL:** Shows the execution of the program and its output:

```
Enter number of students: 5
Enter Student 1 Name: ali
Enter Student 1 Marks: 25
Enter Student 2 Name: ahmed
Enter Student 2 Marks: 25
Enter Student 3 Name: faiq
Enter Student 3 Marks: 25
Enter Student 4 Name: naveed
Enter Student 4 Marks: 25
Enter Student 5 Name: umer
Enter Student 5 Marks: 25
PS D:\Work Space\University\FAST\Classes\PF\FINALS\FILE HANDLING\LAB>
```
- OUTPUT:** Shows the contents of task1.txt:

```
ali : 25
ahmed : 25
faiq : 25
naveed : 25
umer : 25
```

CODE:

```
//Write a C program to read name and marks of n number of students and store them in a file
//Name: Mohsin Ali
//Roll Number: 25P-0545

#include <stdio.h>
#include <string.h>

int main(){

    printf("Name: Mohsin Ali\nRoll Number: 25P-
0545\n_____\n");

    int n;

    printf("Enter number of students: ");

    scanf("%d",&n);
```

```
// int name[n][256];

getchar();
FILE *ptr=fopen("task1.txt","w");

if(ptr==NULL){
    printf("ERROR OPENING! ");
}
else{
    for(int i=0;i<n;i++){
        char name[256];
        int marks;
        printf("Enter Student %d Name: ",i+1);
        scanf("%[^\\n]s",name);
        // fgets(name,sizeof(name),stdin);

        printf("Enter Student %d Marks: ",i+1);
        scanf("%d",&marks);
        getchar();
        fprintf(ptr,"%s : %d\\n",name,marks);
    }
    fclose(ptr);
}
```

Q#2: Write C program to read name and marks of n number of students from and store them in a file. If the file previously exists, add the information to the file.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows files in the LAB folder: .vscode, task1, task1.c, task1.exe, task1.txt, task2, task2.c, task2.exe, task2.txt, task3, task3.bin, task3.c, task4, task4_1.txt, task4_2.txt, task4_1bt, task4_2bt, task4.c, task5, task5.c, task5.txt, task6, task6.c, task6.txt, OUTLINE, and TIMELINE.
- Editor View:** The main editor window displays the content of `task2.c`. The code uses `fopen("task2.txt", "a")` to append data to the file.
- Terminal View:** The terminal window shows the command `PS D:\Work Space\University\FAST\Classes\PF\FINALS\FILE HANDLING\LAB> & 'c:\Users\Ali\vscode\extensions\ms-vscode.cpptools-1.29.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0sf22vsz.trb' '--stdout=Microsoft-MIEngine-Out-oybj5at.c2c' '--stderr=Microsoft-MIEngine-Error-0jfffc13z.c5o' '--pid=Microsoft-MIEngine-Pid-d221a1t4.r1z' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'` followed by the program's output.
- Output:** The terminal output shows the program's execution and the contents of `task2.txt`:

Student Name	Marks
Mohsin	12
Ali	12
faiq	23
naveed	25
umer	23
aftab	12
taha	22
jerry	29
hafa	27
huzaifa	24
Ayan	30
Huzifa	29
Azhan	27

CODE:

```
//Write a C program to read name and marks of n number of students and store them in a file

//Name: Mohsin Ali

//Roll Number: 25P-0545

#include <stdio.h>

#include <string.h>

int main(){

    int n;

    printf("Enter number of students: ");

    scanf("%d",&n);

    getchar();

    FILE *ptr=fopen("task2.txt","a");

    if(ptr==NULL){
```

```
printf("ERROR OPENING! ");

}else{

for(int i=0;i<n;i++){

char name[256];

int marks;

printf("Enter Student %d Name: ",i+1);

scanf("%[^\\n]s",name);

// fgets(name,sizeof(name),stdin);

printf("Enter Student %d Marks: ",i+1);

scanf("%d",&marks);

getchar();

fprintf(ptr,"%s : %d\\n",name,marks);

}

fclose(ptr);

}

}
```

Q#3: C program to write all the members of an array of structures to a file using fwrite(). Read the array from the file and display on the screen.

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows files in the LAB folder: .vscode, task1, task1.c, task1.exe, task1.txt, task2, task2.c, task2.exe, task2.txt, task3, task3.c, task3.bin, task3.exe, task4, task4.c, task4_1.txt, task4_2.txt, task4_f.txt, task4_k.txt, task5, task5.c, task5.txt, task6, task6.c, task6.txt.
- CODE:** The task3.c file contains the following code:

```
task3.c
main()
{
    struct student s[3] = {
        {"ali", 25},
        {"ahmed", 20},
        {"maz", 23}
    };

    //Writing
    FILE *ptr=fopen("task3.bin","wb");

    if(ptr==NULL){
        printf("Error opening file: ");
    }

    Writing Done.....
    Reading.....
    Name: ali : Marks : 25 :
    Name: ahmed : Marks : 20 :
    Name: Maz : Marks : 23 :
}
```
- TERMINAL:** Displays the command-line output of the program execution.

CODE:

```
// C program to write all the members of an array of structures to a file using fwrite(). Read the array
```

```
// from the file and display on the screen.
```

```
//Name: Mohsin Ali
```

```
//Roll Number: 25P-0545
```

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

```
#include <string.h>
```

```
struct student{
```

```
    char name[256];
```

```
    int marks;
```

```
};

int main(){
```

```
    struct student s[3]={  
        {"ali",25},  
        {"ahmed",20},  
        {"Maz",23}  
    };
```

```
//Writing
```

```
FILE *ptr=fopen("task3.bin","wb");  
  
if(ptr==NULL){  
    printf("Error Opening File: ");  
}
```

```
fwrite(s,sizeof(struct student),3,ptr);  
printf("Writing Done.....\n");  
fclose(ptr);
```

```
//Reading
```

```
printf("Reading.....\n");  
ptr=fopen("task3.bin","rb");  
if(ptr==NULL){  
    printf("Error Opening File: ");
```

```
}

struct student s_read[3];

fread(s_read,sizeof(struct student),3,ptr);

//Display

for(int i=0;i<3;i++){
    printf("Name: %s : ",s_read[i].name);
    printf("Marks : %d : \n",s_read[i].marks);

}

}
```

Q#4: Write a C program to create 2 text files and store some text inside them. Then read these 2 files into the program and merge the text into a 3rd text file.

```
C:\task4> main()
1 // Write a C program to create 2 text files and store some text inside them. Then read these 2 files into
2 // memory and merge the text into a 3rd text file.
3 //Name: Mohsin Ali
4 //Roll Number: 2SP-0545
5
6 #include <stdio.h>
7
8 int main()
9 {
10     FILE *ptr=fopen("task4_1.txt","a");
11
12     fputs("Hello World.",ptr);
13     fclose(ptr);
14
15     FILE *ptr1=fopen("task4_2.txt","a");
16
17     fputs("My Name Is Ali, I am a CS Student.",ptr1);
18
19     fclose(ptr1);
20
21     FILE *ptr2=fopen("task4_3.txt","a");
22     ptr1=fopen("task4_2.txt","r");
23     ptr2=fopen("task4_1.txt","r");
24
25
26     char ch;
27
28     while((ch=getchar(ptr))!=EOF)
29     {
30         fputchar(ch,ptr2);
31     }
32     while((ch=getchar(ptr))!=EOF)
33     {
34         fputchar(ch,ptr2);
35     }
36
37     fclose(ptr);
38     fclose(ptr2);
39     fclose(ptr1);
40
41     printf("Program Run Successfull ");
42 }
```

CODE:

// //Write a C program to create 2 text files and store some text inside them. Then read these 2 files into

// the program and merge the text into a 3rd text file.

//Name: Mohsin Ali

//Roll Number: 25P-0545

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

```
int main(){
```

FILE *ptr

```
fputs("Hello World. ",ptr);
```

`fclose(ptr);`

FILE *ptr1=

```
fputs("My Name is Ali. I am a CS Student");
```

fclose(ptr1);

```
FILE *ptr2=fopen("task4_f.txt","w");
ptr1=fopen("task4_2.txt","r");
ptr=fopen("task4_1.txt","r");
char ch;
while((ch=fgetc(ptr))!=EOF){
    fputc(ch,ptr2);
}
while((ch=fgetc(ptr1))!=EOF){
    fputc(ch,ptr2);
}
fclose(ptr);
fclose(ptr2);
fclose(ptr1);
printf("Program Run Successfull! ");
}
```

Q#5: Write C Program that Reads a Text File and Counts the Number of Times a Certain Letter Appears in the Text File

The screenshot shows the Visual Studio Code interface with two tabs open: 'task5.c' and 'task5.txt'. The 'task5.c' tab contains the C program code, and the 'task5.txt' tab contains the test file content.

```
task5.c
1 //Write C Program that Reads a Text File and Counts the Number of Times a Certain Letter
2 //Appears in the Text File
3 //Name: Mohsin Ali
4 //Roll Number: 25P-0545
5 #include <stdio.h>
6 int main(){
7     FILE *ptr=fopen("task5.txt","r");
8
9     if(ptr==NULL){
10         printf("Error Opneing the file: ");
11         return 0;
12     }
13     char x;
14     printf("Enter Character to look for: ");
15     scanf(" %c", &x);
16
17     char ch;
18
19     int count=0;
20     while(ch=fgetc(ptr))!=EOF{
21
22         if(ch==x){
23             count++;
24         }
25     }
26
27     fclose(ptr);
28
29     printf("Character %c found %d number of times: ",x,count);
30 }
```

```
task5.txt
1 Hello, This is a test file for task 5 where it will look for a number of a specific character.
2 so I am writing these lines to look for a specific character.
```

CODE:

```
// //Write C Program that Reads a Text File and Counts the Number of Times a Certain Letter
//Appears
```

```
// in the Text File
```

```
//Name: Mohsin Ali
```

```
//Roll Number: 25P-0545
```

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

```
int main(){
```

```
FILE *ptr=fopen("task5.txt","r");
```

```
if(ptr==NULL){
```

```
    printf("Error Opneing the file: ");
```

```
    return 0;
```

```
}
```

```
char x;
```

```
printf("Enter Character to look for: ");

scanf(" %c",&x);

char ch;

int count=0;

while((ch=fgetc(ptr))!=EOF){

    if(ch==x){

        count++;

    }

}

fclose(ptr);

printf("Character %c found %d number of times: ",x,count);

}
```

Q#6: Consider a String entered by the user which he wants to encrypt and then decrypt the information. Perform this functionality via the filling technique by first inserting the encrypted text in the file then read the encrypted text to decode it back to string.

```

task6.c
1 // Consider a String entered by the user which he wants to encrypt and then decrypt the
2 // information. Perform this functionality via the filling technique by first inserting the
3 // encrypted text in the file then read the encrypted text to decode it back to string.
4 //Name: Mohsin Ali
5 //Roll Number: 25P-0545
6 #include <stdio.h>
7 #include <string.h>
8
9 int main(){
10
11     char word[256];
12
13     printf("Enter Word: ");
14
15     fgets(word,256,stdin);
16
17     word[strlen(word)] = '\0';
18
19     int ln(strlen(word));
20
21     for(int i=0;word[i]!='\0';i++){
22         word[i]=i+1;
23     }
24     printf("\nEncrypted text: \n");
25     for(int i=0;i<ln;i++){
26         printf("%c",word[i]);
27     }
28
29     FILE *ptr=fopen("task6.txt","w");
30
31     fputs(word,ptr);
32
33 }

```

task6.txt

task6.txt

LAB

task1

task1.c

task1.exe

task1.txt

task2

task2.c

task2.exe

task2.txt

task3

task3.c

task3.exe

task3.txt

task4

task4.c

task4.exe

task4_1.txt

task4_2.txt

task4_f.txt

task4_c

task5

task5.c

task5.exe

task5.txt

task6

task6.c

task6.exe

task6.txt

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Work Space\University\FAST\Classes\PfF\FINALS\FILE HANDLING\LAB> & "c:\Users\Ali1\vscode\extensions\ms-vscode.cpptools-1.29.0\win32-x64\debugAdapters\vbin\WindowsDebugLauncher.exe" <-stdin=>Microsoft-MIEngine-In-aqeljtws3.vut <-stdout=>Microsoft-MIEngine-Out-djelokump.w4n <-stderr=>Microsoft-MIEngine-Error-hieitlw.jjs <-pid=>Microsoft-MIEngine-PId-3y9qm1ny.hel <-dbgExe=C:\msys64\usr\bin\gdb.exe" "-interpreter=mi"

Enter Word: ABC

Encrypted text:

BDF

Decrypted Text:

ABC

PS D:\Work Space\University\FAST\Classes\PfF\FINALS\FILE HANDLING\LAB> []

CODE:

```

// Consider a String entered by the user which he wants to encrypt and then decrypt the
information.

// Perform this functionality via the filling technique by first inserting the encrypted text in the
file

// then read the encrypted text to decode it back to string.

//Name: Mohsin Ali

//Roll Number: 25P-0545

#include <stdio.h>

#include <string.h>

int main(){

char word[256];

printf("Enter Word: ");


```

```
fgets(word,256,stdin);
```

```
word[strcspn(word, "\n")] = '\0';
```

```
int ln=strlen(word);
```

```
for(int i=0;word[i]!='\0';i++){
```

```
    word[i]+=(i+1);
```

```
}
```

```
printf("\nEncrypted text: \n");
```

```
for(int i=0;i<ln;i++){
```

```
    printf("%c",word[i]);
```

```
}
```

```
FILE *ptr=fopen("task6.txt","w");
```

```
fputs(word,ptr);
```

```
fclose(ptr);
```

```
ptr=fopen("task6.txt","r");
```

```
for(int i=0;word[i]!='\0';i++){
```

```
    word[i]=-(i+1);
```

```
}

printf("\nDecrypted Text: \n");

printf("%s",word);

}
```