

***NAME : Himanshu Dixit***  
***ENROLL NO. : B64178***  
***BATCH : B10***

**SOFTWARE DEVELOPMENT FUNDAMENTAL LAB-I(15B17CI171)**  
**Assignment Sheet (WEEK-11 PHASE-2)**  
**Lab A**

**1. Write a program in C to create and store information in a text file.**

*Sample input:*

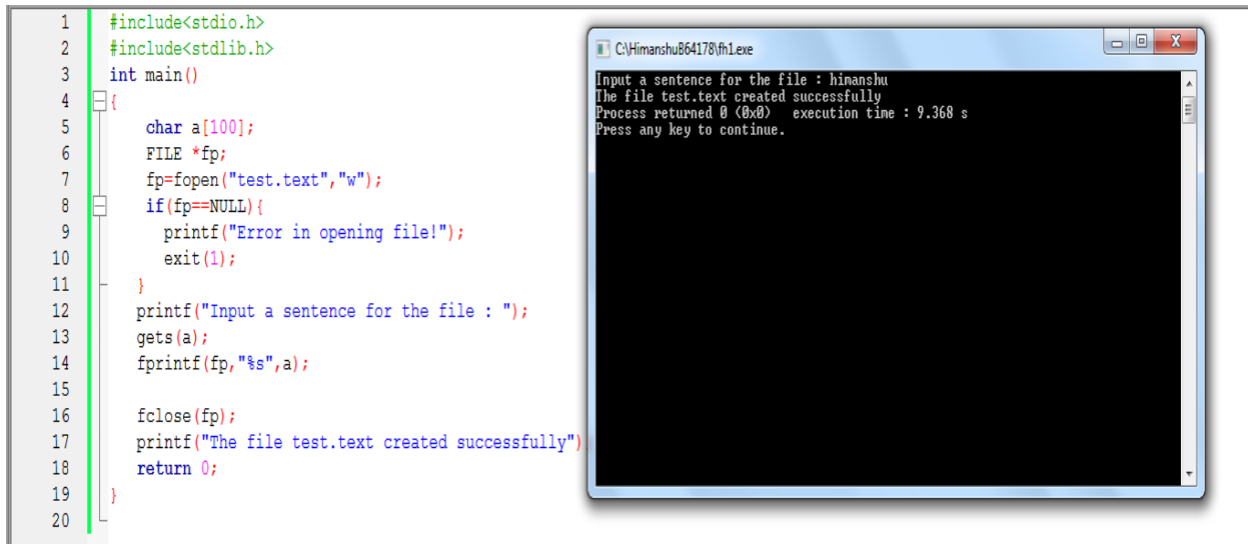
*Input a sentence for the file : This is the content of the file test.txt.*

*Expected Output:*

*The file test.txt created successfully...!!*

**Solution:**

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char a[100];
    FILE *fp;
    fp=fopen("test.text","w");
    if(fp==NULL){
        printf("Error in opening file!");
        exit(1);
    }
    printf("Input a sentence for the file : ");
    gets(a);
    fprintf(fp,"%s",a);
    fclose(fp);
    printf("The file test.text created successfully");
    return 0;
}
```

The image shows a C program on the left and its execution output on the right. The program is a file creation utility. It includes `<stdio.h>` and `<stdlib.h>`, and has a `main` function. It declares a character array `a` of size 100 and a file pointer `fp`. It attempts to open a file named `test.text` in write mode. If the file is not created (which is not the case here), it prints an error and exits. Since the file is created successfully, it prompts the user for a sentence, reads it using `gets`, and writes it to the file using `fprintf`. Finally, it closes the file and prints a success message before returning 0. The execution window on the right shows the program's output: it prompts for a sentence, confirms the file was created successfully, shows the return code as 0 and execution time as 9.368 seconds, and then waits for a key press to continue.

**2. Write a program in C to read an existing file.**

*Test Data :*

*Input the file name to be opened : test.txt*

*Expected Output :*

*The content of the file test.txt is :*

*This is the content of the file test.txt.*

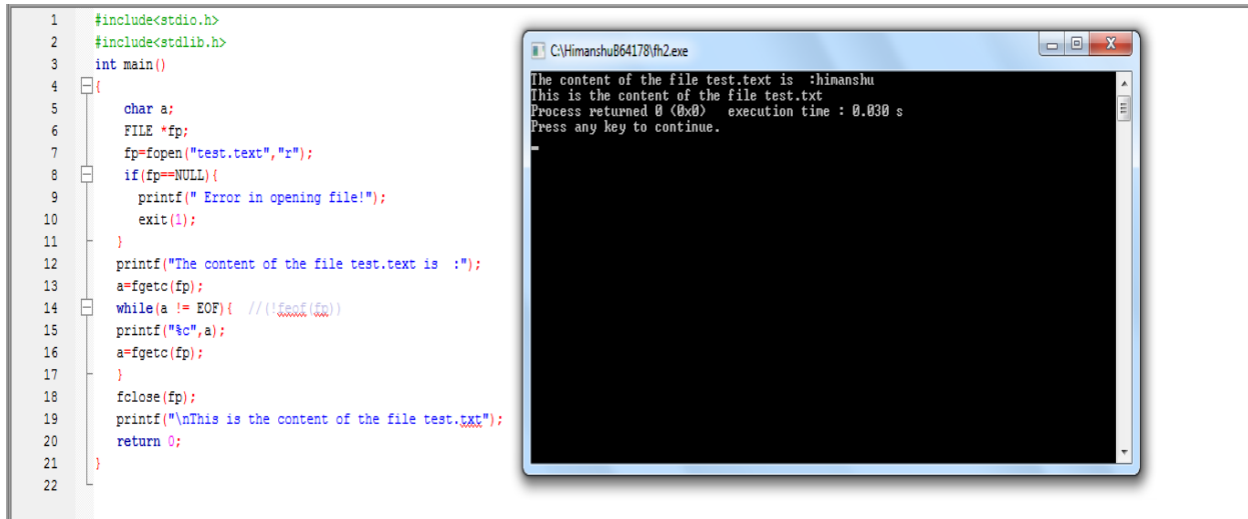
**Solution:**

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char a;
    FILE *fp;
    fp=fopen("test.text","r");
    if(fp==NULL){
        printf(" Error in opening file!");
        exit(1);
    }
    printf("The content of the file test.text is :");
    a=fgetc(fp);
    while(a != EOF){ //(feof(fp))
        printf("%c",a);
```

```

a=fgetc(fp);
}
fclose(fp);
printf("\nThis is the content of the file test.txt");
return 0;
}

```



**3. Write a program in C to write multiple lines in a text file.**

*Test Data :*

*Input the number of lines to be written : 4*

*:: The lines are ::*

*test line 1*

*test line 2*

*test line 3*

*test line 4*

*Expected Output :*

*The content of the file test.txt is :*

*test line 1*

*test line 2*

*test line 3*

*test line 4*

**Solution:**

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

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

```
int main ()
```

```

{
    char a[100],b;
    FILE *fp;
    fp=fopen("test.text","w");
    char fname[20]="test.txt";
    int n;
    printf("Input the number of lines to be written : ");
    scanf("%d", &n);
    printf("The lines are :\n");
    for(int i = 0; i < n+1; i++)
    {
        fgets(a, sizeof(a), stdin);
        fputs(a, fp);
    }
    fclose (fp);
    fp=fopen("test.text","r");
    printf("\nThe content of the file %s is :",fname);
    b = fgetc(fp);
    while (b != EOF)
    {
        printf ("%c", b);
        b = fgetc(fp);
    }
    fclose (fp);
    return 0;
}

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main ()
4  {
5      char a[100],b;
6      FILE *fp;
7      fp=fopen("test.text","w");
8      char fname[20]="test.txt";
9      int n;
10     printf("Input the number of lines to be written : ");
11     scanf("%d", &n);
12     printf("The lines are :\n");
13     for(int i = 0; i < n+1; i++)
14     {
15         fgets(a, sizeof(a), stdin);
16         fputs(a, fp);
17     }
18     fclose (fp);
19     fp=fopen("test.text","r");
20     printf("\nThe content of the file %s is :",fname);
21     b = fgetc(fp);
22     while (b != EOF)
23     {
24         printf ("%c", b);
25         b = fgetc(fp);
26     }
27     fclose (fp);
28     return 0;
29 }
30

```

C:\Himanshu864178\fn3.exe  
 Input the number of lines to be written : 2  
 The lines are :  
 hinanshu  
 harsh  
 The content of the file test.txt is :  
 hinanshu  
 harsh  
 Process returned 0 (0x0) execution time : 17.379 s  
 Press any key to continue.

4. Write a program in C to find the number of lines in a text file.

Enter file name: abc.txt

There are 43 lines in the file

**Solution:**

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

```
int main()
```

```
{
```

```
    FILE *fp;
```

```
    int c = 0;
```

```
    char filename[20], chr;
```

```
    printf("Enter file name: ");
```

```
    scanf("%s", filename);
```

```
    fp = fopen(filename, "r");
```

```
    chr = getc(fp);
```

```
    while (chr != EOF)
```

```
    {
```

```
        if (chr == '\n')
```

```
        {
```

```
            c++;
```

```
        }
```

```
        chr = getc(fp);
```

```
    }
```

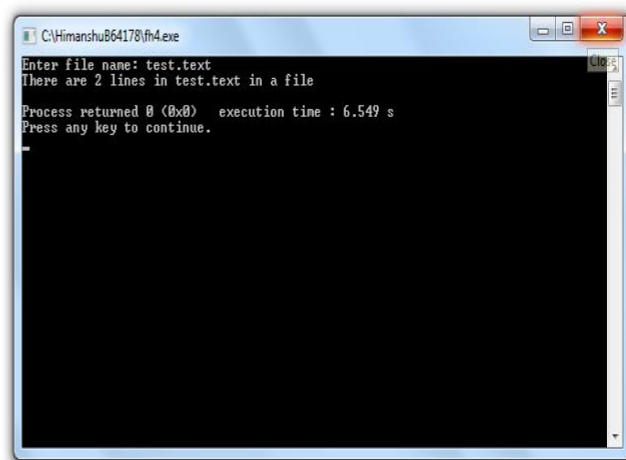
```
    fclose(fp);
```

```
    printf("There are %d lines in %s in a file\n", c, filename);
```

```
    return 0;
```

```
}
```

```
1  #include <stdio.h>
2  int main()
3  {
4      FILE *fp;
5      int c = 0;
6      char filename[20], chr;
7      printf("Enter file name: ");
8      scanf("%s", filename);
9      fp = fopen(filename, "r");
10     chr = getc(fp);
11     while (chr != EOF)
12     {
13         if (chr == '\n')
14         {
15             c++;
16         }
17         chr = getc(fp);
18     }
19     fclose(fp);
20     printf("There are %d lines in %s in a file\n", c, filename);
21     return 0;
22 }
23
```



```
C:\Himanshu\B64178\fin4.exe
Enter file name: test.txt
There are 2 lines in test.txt in a file
Process returned 0 (0x0) execution time : 6.549 s
Press any key to continue.
```

**5. Write a C Program to append the content of file at the end of another.**

*Enter name of first file a.txt*

*Enter name of second file b.txt*

*Enter name to store merged file merge.txt*

*Two files merged merge.txt successfully.*

**Solution:**

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp1, *fp2, *ft;
    char ch, file1[20], file2[20], file3[20];
    printf("Enter name of first file ");
    gets(file1);
    printf("Enter name of second file ");
    gets(file2);
    printf("Enter name to store merged file ");
    gets(file3);
    fp1 = fopen(file1, "r");
    fp2 = fopen(file2, "r");
    if (fp1 == NULL || fp2 == NULL)
    {
        printf("Error in opening file!");
        exit(1);
    }
    ft = fopen(file3, "w");
    if (ft == NULL)
    {
        printf("Error in opening file!");
        exit(1);
    }
    while ((ch = fgetc(fp1)) != EOF){
        fputc(ch, ft);
    }
    while ((ch = fgetc(fp2)) != EOF){
```

```

        fputc(ch, ft);
    }
    printf("Two files merged %s successfully.", file3);
    fclose(fp1);
    fclose(fp2);
    fclose(ft);
return 0;
}

```

The image shows a C program on the left and its execution output in a terminal window on the right.

**Program Code:**

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      FILE *fp1, *fp2, *ft;
6      char ch, file1[20], file2[20], file3[20];
7      printf("Enter name of first file ");
8      gets(file1);
9      printf("Enter name of second file ");
10     gets(file2);
11     printf("Enter name to store merged file ");
12     gets(file3);
13     fp1 = fopen(file1, "r");
14     fp2 = fopen(file2, "r");
15     if (fp1 == NULL || fp2 == NULL)
16     {
17         printf("Error in opening file!");
18         exit(1);
19     }
20     ft = fopen(file3, "w");
21     if (ft == NULL)
22     {
23         printf("Error in opening file!");
24         exit(1);
25     }
26     while ((ch = fgetc(fp1)) != EOF){
27         fputc(ch, ft);
28     }
29     while ((ch = fgetc(fp2)) != EOF){
30         fputc(ch, ft);
31     }
32     printf("Two files merged %s successfully.", file3);
33     fclose(fp1);
34     fclose(fp2);
35     fclose(ft);
36     return 0;
37 }

```

**Terminal Output:**

```

C:\Himanshu864178\fh5.exe
Enter name of first file test.txt
Enter name of second file abc.txt
Enter name to store merged file temp.txt
Two files merged temp.txt successfully.
Process returned 0 (0x0)   execution time : 21.316 s
Press any key to continue.

```