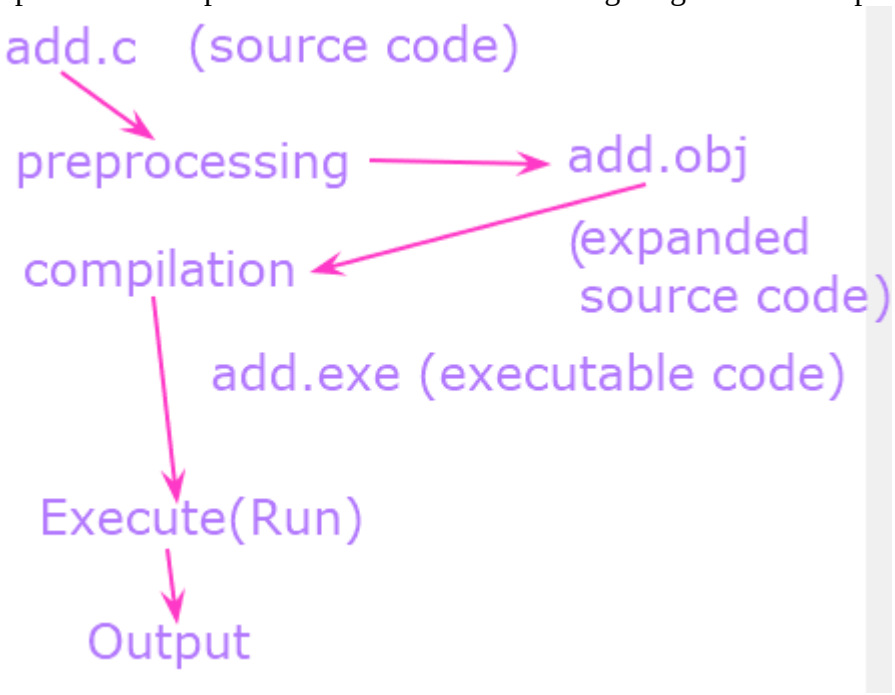
********** Unit - III **********
Preprocessor
Files

---------- Preprocessor ----------
is used as preprocessor.

The preprocessor will process the source code before giving it to the compiler.



The following are the various preprocessor directives used in c language
1. Macro (#define)
2. #include
3. #ifdef
4. #ifndef
5. #undef

1. Macro :
We use the macros to define the constants in the program.

We can define the macros using #define preprocessor directive

syntax:
#define Macroname MacroReplacement

In general the macro names will be defined in upper case letters

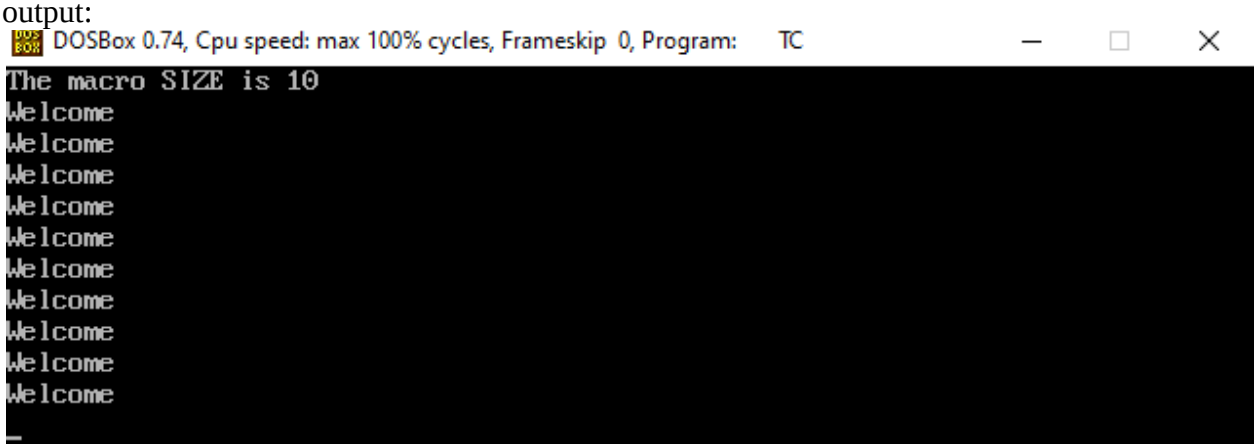
#define SIZE 10

Write a program to demonstrate the creation and usage of a macro

prepro1.c

```
#include<stdio.h>
#define SIZE 10
#define WEL printf("Welcome\n");
main()
{
        int i;
        clrscr();
        printf("The macro SIZE is %d\n",SIZE);
        for(i=0;i<SIZE;i++)
        {
                WEL
        }
        return 0;
}
```

output:



2. #include:
#include preprocessor directive is used to include header libraries in the program or to link the other c programs with the current program.

ex:

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

#include<prog2.c>
```

3. #ifdef:
This directive will verifies whether a macro is defined or not

syntax:

#ifdef Macroname
statements
#else
statements
#endif

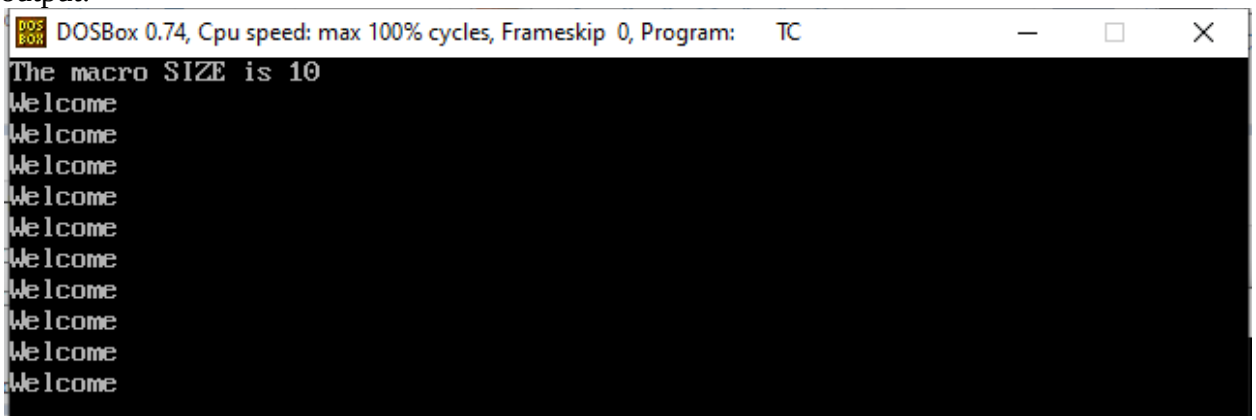
If the macro is defined properly then it executes the statements written after #ifdef otherwise it executes the statements written after #else

Write a program to demonstrate the #ifdef for verifying whether macro is defined or not.

prepro2.c

```
#define SIZE 10
#define WEL printf("Welcome\n");
main()
{
        int i;
        clrscr();
        printf("The macro SIZE is %d\n",SIZE);

        #ifdef SIZE

        for(i=0;i<SIZE;i++)
        {
                WEL
        }
        #else
        printf("The Macro SIZE is not defined\n");
        #endif

        return 0;
}
```

output:



4. #ifndef:
#ifndef will checks whether a macro is not defined. If macro is not defined then it executes the statements written after #ifndef otherwise it executes the statements written after #else

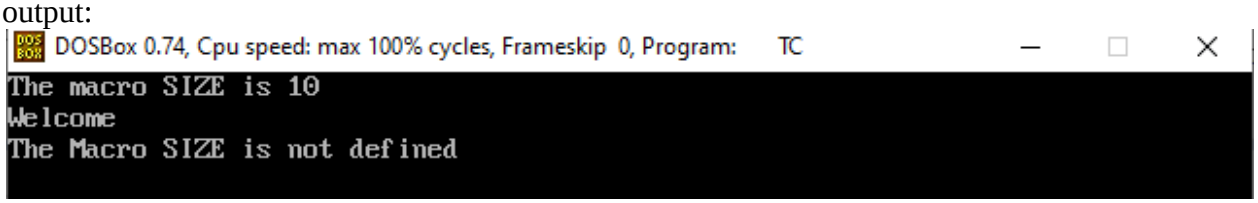
syntax:
#ifndef macroname
statments
#else
statements
#endif

Write a program to demonstrate the #ifndef for checking whether a macro is not defined or not.

prepro3.c

```
#include<stdio.h>
#define SIZE 10
#define WEL printf("Welcome\n");
main()
{
	int i;
	clrscr();
	printf("The macro SIZE is %d\n",SIZE);

	#undef SIZE

	#ifndef SIZE
		WEL
		printf("The Macro SIZE is not defined\n");
	#else
		for(i=0;i<SIZE;i++)
		{
			WEL
		}
	#endif

	return 0;
}
```

output:

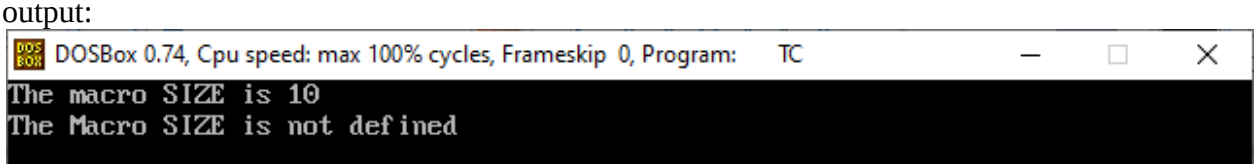


5. #undef:
#undef will undefines the defined macros

prepro4.c

```
#include<stdio.h>
#define SIZE 10
#define WEL printf("Welcome\n");
main()
{
        int i;
        clrscr();
        printf("The macro SIZE is %d\n",SIZE);

        #undef SIZE

        #ifdef SIZE

        for(i=0;i<SIZE;i++)
        {
                WEL
        }
        #else
        printf("The Macro SIZE is not defined\n");
        #endif

        return 0;
}
```

output:



********** Files **********

File : A file is a collection of data stores on the disk perminantly.

We can create and access the files using c program in the following steps.

Step 1: Create a file pointer
Step 2: Open the file
Step 3: Access the file
Step 4: Close the file

Step 1: Create a file pointer:

We can create a file pointer using FILE as follows.

syntax:
FILE *filepointername;

Ex:
FILE *fp;

File Pointer stores the information related to the file like filename, file access mode.

Step 2: Open the file:

Once we create a file pointer we need to open the file for accessing.
We can open the file using fopen() as follows

syntax:
filepointername=fopen("FileName","FileAccessMode");

File Access Mode specifies the purpose we are opening the file like reading,writing and appending.

File Access Mode	Text Files	Binary Files
read	r	rb
write	w	wb
append	a	ab
read/write	r+	rb+
write/read	w+	wb+
append plus	a+	ab+

read(r) : We open the file to read the contents from the file.
write(w) : We open the file to write(saving) the contents to the file.

append(a) : We can open an existing file and append(add) some new contents to the file
read/write(r+) : We open an existing file, read the contents of it and write the new the contents to the file.
write/read(w+) : We open an a file, write the contents to the file and read the contents from the file.
append plus(a+) : We can open a file, append the contents to the file and read the contents from the file.

Ex:
fp=fopen("file1.txt","w");

fopen() will returns the filename and fileaccessmode to the file pointer if the file opened successfully, otherwise it returns NULL.

Step 3: Access the file:

After opening the file we can access the contents of a file using file input/output functions

Accessing the file means saving or appending the data to the file and reading the data from the file.

Step 4: Close the file :

After accesing the file contents we must close the file.
We can close the file using fclose() as follows.

syntax:
fclose(filepointername);

Ex:
fclose(fp);

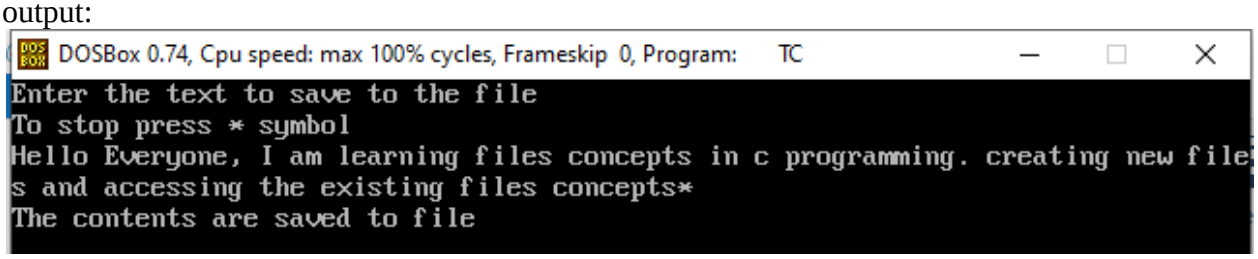
Write a program to create file in write mode and save the contents into it.

file1.c

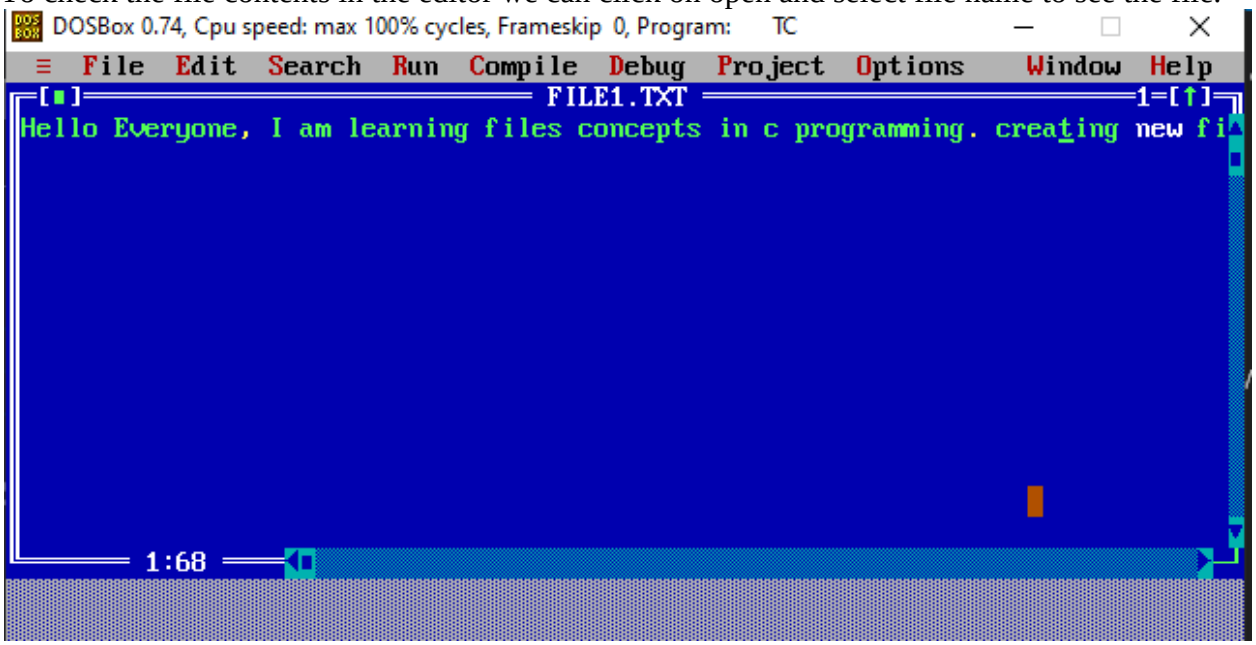
```
#include<stdio.h>
main()
{
        FILE *fp;
        char ch;
        clrscr();
        fp=fopen("file1.txt","w");
        if(fp==NULL)
        {
                exit(0);
```

```
        }
        printf("Enter the text to save to the file\n");
        printf("To stop press * symbol\n");
        while((ch=getchar())!='*')
        {
                putc(ch,fp);
        }
        printf("The contents are saved to file\n");
        fclose(fp);
        return 0;
}
```

output:



To check the file contents in the editor we can click on open and select file name to see the file.



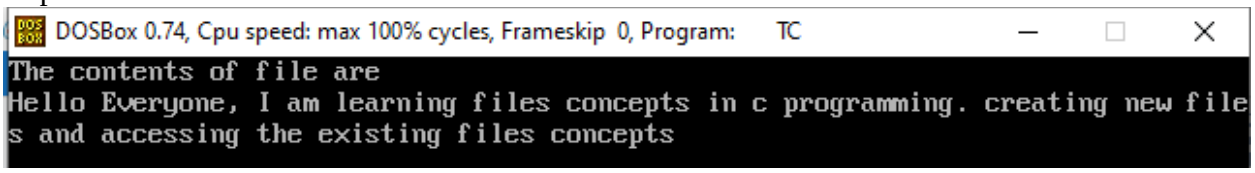
Write a program to open a file in read mode, read the contents from the file and display on the console.

file2.c

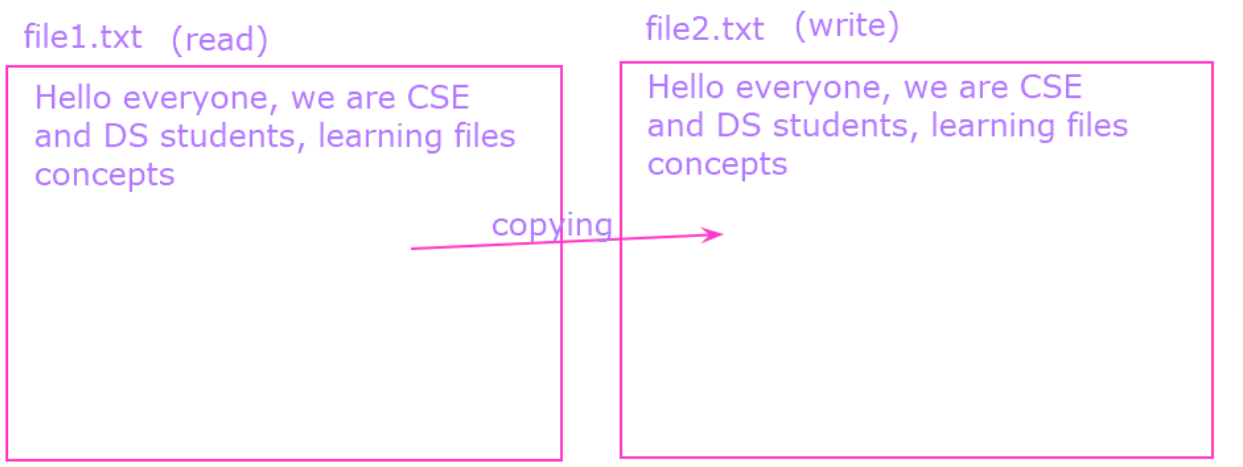
```
#include<stdio.h>
main()
```

```
{
        FILE *fp;
        char ch;
        clrscr();
        fp=fopen("file1.txt","r");
        if(fp==NULL)
        {
                printf("File not opened successfully\n");
                exit(0);
        }
        printf("The contents of file are\n");
        while((ch=getc(fp))!=EOF)
        {
                putchar(ch);
        }
        fclose(fp);
        return 0;
}
```

output:



Write a program to copy the contents of one file to other file(from file1.txt to file2.txt)

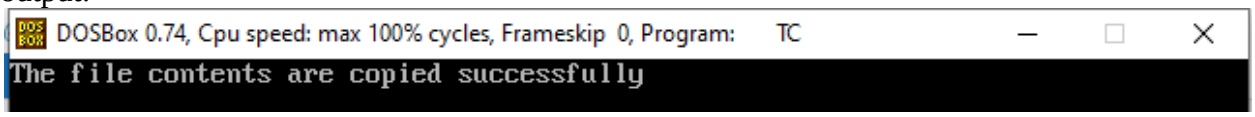


file3.c

```
#include<stdio.h>
main()
{
```

```
        FILE *fpr,*fpw;
        char ch;
        clrscr();
        fpr=fopen("file1.txt","r");
        if(fpr==NULL)
        {
                printf("File not opened successfully\n");
                exit(0);
        }
        fpw=fopen("file2.txt","w");
        if(fpw==NULL)
        {
                printf("File not created successfully\n");
                exit(0);
        }
        while((ch=getc(fpr))!=EOF)
        {
                putc(ch,fpw);
        }
        printf("The file contents are copied successfully\n");
        fclose(fpr);
        fclose(fpw);
        return 0;
}
```

output:



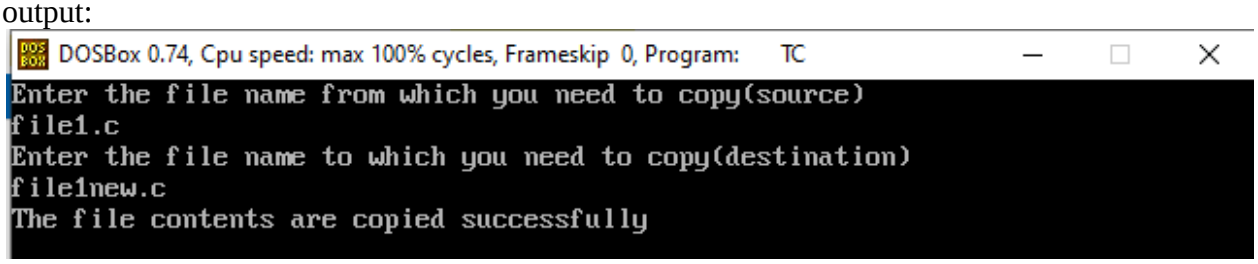
Write a program to read source and destination filenames from keyboard and copy the contents from the source file to destination file.

file4.c

```
#include<stdio.h>
main()
{
        FILE *fpr,*fpw;
        char ch;
        char fname1[15],fname2[15];
        clrscr();
        printf("Enter the file name from which you need to copy(source)\n");
        gets(fname1);
        printf("Enter the file name to which you need to copy(destination)\n");
        gets(fname2);
        fpr=fopen(fname1,"r");
```

```
	if(fpr==NULL)
	{
		printf("File not opened successfully\n");
		exit(0);
	}
	fpw=fopen(fname2,"w");
	if(fpw==NULL)
	{
		printf("File not created successfully\n");
		exit(0);
	}
	while((ch=getc(fpr))!=EOF)
	{
		putc(ch,fpw);
	}
	printf("The file contents are copied successfully\n");
	fclose(fpr);
	fclose(fpw);
	return 0;
}
```

output:



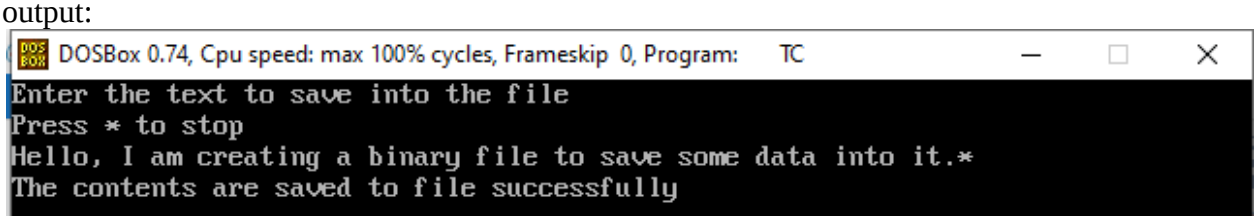
Write a program to create a binary file in write mode and save the contents the file.

file5.c

```
#include<stdio.h>
main()
{
	FILE *fp;
	char ch;
	clrscr();
	fp=fopen("bfile1.dat","wb");
	if(fp==NULL)
	{
		printf("File not created successfully\n");
		exit(0);
	}
	printf("Enter the text to save into the file\n");
```

```
        printf("Press * to stop\n");
        while((ch=getchar())!='*')
        {
                putc(ch,fp);
        }
        printf("The contents are saved to file successfully\n");
        fclose(fp);
        return 0;
}
```

output:

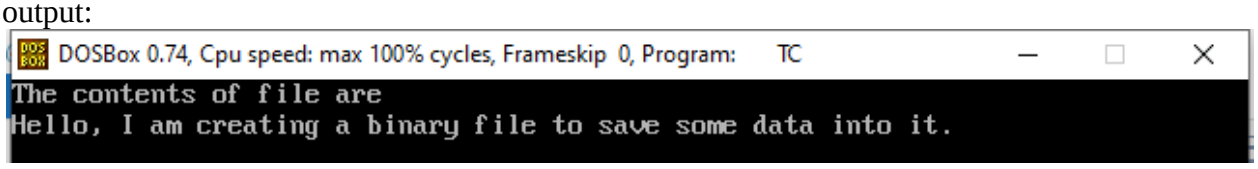


Write a program to open a binary file in read mode and display the contents on the console.

file6.c

```
#include<stdio.h>
main()
{
        FILE *fp;
        char ch;
        clrscr();
        fp=fopen("bfile1.dat","rb");
        if(fp==NULL)
        {
                printf("File not opened successfully\n");
                exit(0);
        }
        printf("The contents of file are\n");
        while((ch=getc(fp))!=EOF)
        {
                putchar(ch);
        }
        fclose(fp);
        return 0;
}
```

output:

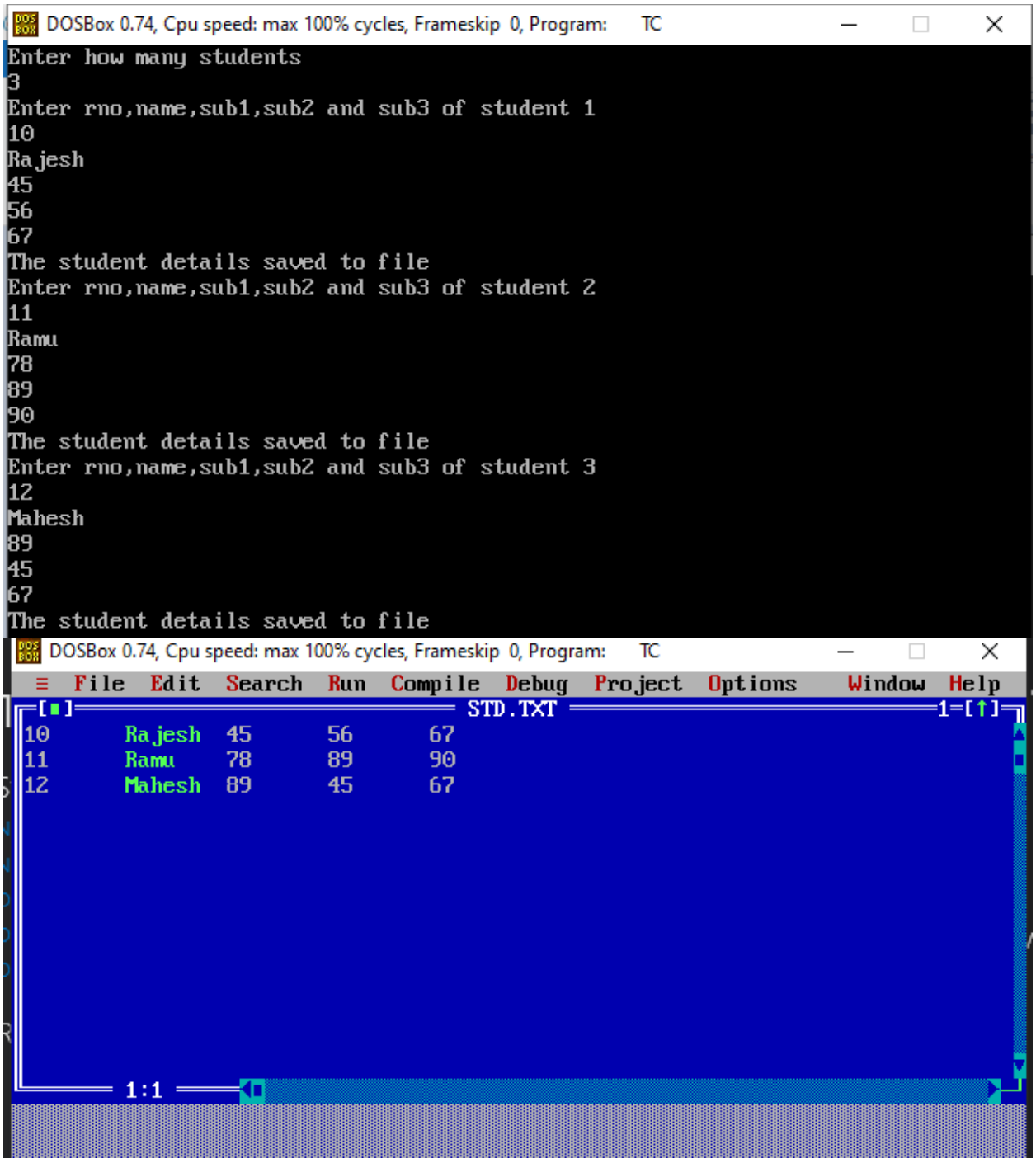


Write a program to create a file to save the details of a student like rno,name, sub1,sub2 ans sub3 marks.(use fprintf() function)

file7.c

```
#include<stdio.h>
main()
{
	FILE *fp;
	int rno,sub1,sub2,sub3;
	char name[15];
	int n,i;
	clrscr();
	fp=fopen("std.txt","w");
	if(fp==NULL)
	{
		printf("File not created successfully\n");
		exit(0);
	}
	printf("Enter how many students\n");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("Enter rno,name,sub1,sub2 and sub3 of studetn %d\n",i+1);
		scanf("%d",&rno);
		scanf("%s",name);
		scanf("%d%d%d",&sub1,&sub2,&sub3);
		fprintf(fp,"%d\t%s\t%d\t%d\t%d\n",rno,name,sub1,sub2,sub3);
		printf("The student details saved to file\n");
	}
	fclose(fp);
	return 0;
}
```

output:



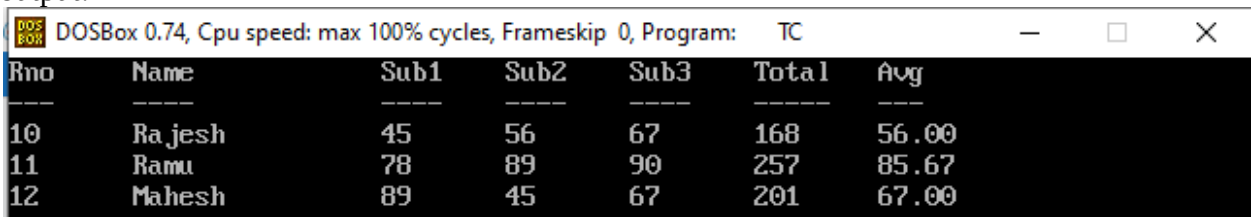
Write a program to read the contents of std.txt file and display them in a tabular by calculating the total marks and avg also.(using fscanf() function)

Rno	Name	Sub1	Sub2	Sub3	Total	Avg

file8.c

```
#include<stdio.h>
main()
{
	FILE *fp;
	int rno,sub1,sub2,sub3;
	char name[15];
	int total;
	float avg;
	clrscr();
	fp=fopen("std.txt","r");
	if(fp==NULL)
	{
		printf("File not opened successfully\n");
		exit(0);
	}
	printf("Rno\tName\t\tSub1\tSub2\tSub3\tTotal\tAvg\n");
	printf("---\t----\t\t----\t----\t----\t-----\t---\n");
	while(!feof(fp))
	{
		fscanf(fp,"%d%s%d%d%d",&rno,name,&sub1,&sub2,&sub3);
		total=sub1+sub2+sub3;
		avg=total/3.0;
		printf("%d\t%s\t\t%d\t%d\t%d\t%d\t%.2f\n",rno,name,sub1,sub2,sub3,total,avg);
	}
	fclose(fp);
	return 0;
}
```

output:



HW1:
Write a program to create an employee file and save details of employee eno,ename and salary(using fprintf() function)(emp.txt)

HW2:
Write a program to read the contents of employee file and display them in the following format.

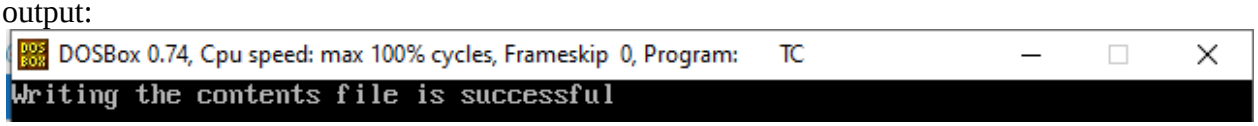
```
Eno	EName	Salary
---	-----	------
```

Write a program to create a binary file and write the data to the file.
(using fwrite() function)

file9.c

```
#include<stdio.h>
main()
{
	int a[5]={10,20,30,40,50};
	FILE *fp;
	int n;
	clrscr();
	fp=fopen("bin1.dat","wb");
	if(fp==NULL)
	{
		printf("File not created successfully\n");
		exit(0);
	}
	n=fwrite(a,sizeof(int),5,fp);
	if(n==5)
	{
		printf("Writing the contents file is successful\n");
	}
	else
	{
		printf("Writing is failed\n");
	}
	return 0;
}
```

output:



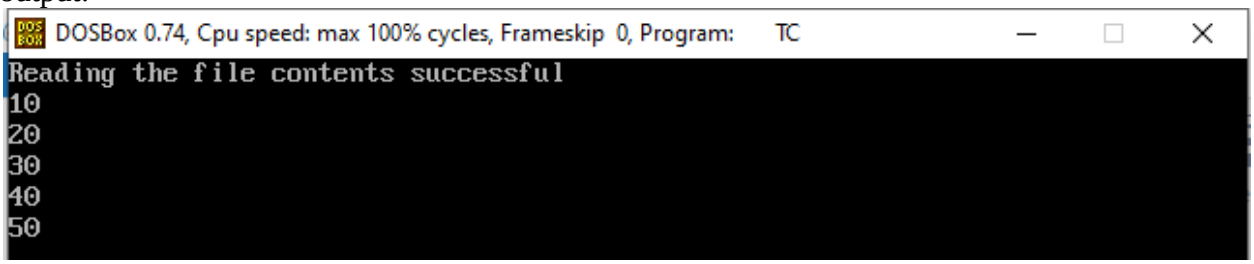
Write a program to read the contents of a binary file and display them on the console(using fread() function)

file10.c

```
#include<stdio.h>
main()
{
	int a[5],i;
	FILE *fp;
	int n;
	clrscr();
	fp=fopen("bin1.dat","rb");
	if(fp==NULL)
```

```
        {
                printf("File not opened successfully\n");
                exit(0);
        }
        n=fread(a,sizeof(int),5,fp);
        if(n==5)
        {
                printf("Reading the file contents successful\n");
                for(i=0;i<n;i++)
                {
                        printf("%d\n",a[i]
        }
        else
        {
                printf("Reading is failed\n");
        }
        return 0;
}
```

output:



Sequential Access Files: In this we can access the contents from the begining of the file to till the end of the file.

Random Access Files: We can access the contents of file from the random locations of the file.

We can use the following functions to access the contents randomly from the file.

1. fseek()
2. ftell()
3. rewind()

1. fseek():
This function will place the cursor at the specified position.

syntax:

int fseek(FILE *filepointer,long int offset,int position);

position constants

SEEK_SET - 0 : Begining of the file
SEEK_CUR - 1 : Current position of the cursor
SEEK_END - 2 : End of the file

Example:
1. fseek(fp,0,SEEK_SET) : It places the cursor at the begining of the file.
2. fseek(fp,10,SEEK_SET) : It places the cursor after 10 characters from the begining of the file.
3. fseek(fp,0,SEEK_CUR) : It places the cursor at the current position of the cursor.
4. fseek(fp,10,SEEK_CUR) : It places the cursor after 10 characters from the current position.
5. fseek(fp,-10,SEEK_CUR) : It places the cursor before 10 characters from the current position.
6. fseek(fp,0,SEEK_END) : It places the cursor at the end of the file.
7. fseek(fp,-10,SEEK_END) : It places the cursor before 10 characters from the end of the file.

2. ftell():
It tells the current position of cursor.

syntax:

long int ftell(FILE *filepointer);

Ex:
long int pos;
pos=ftell(fp);

3. rewind():
It repositions(replaces) the cursor at the begining of the file.

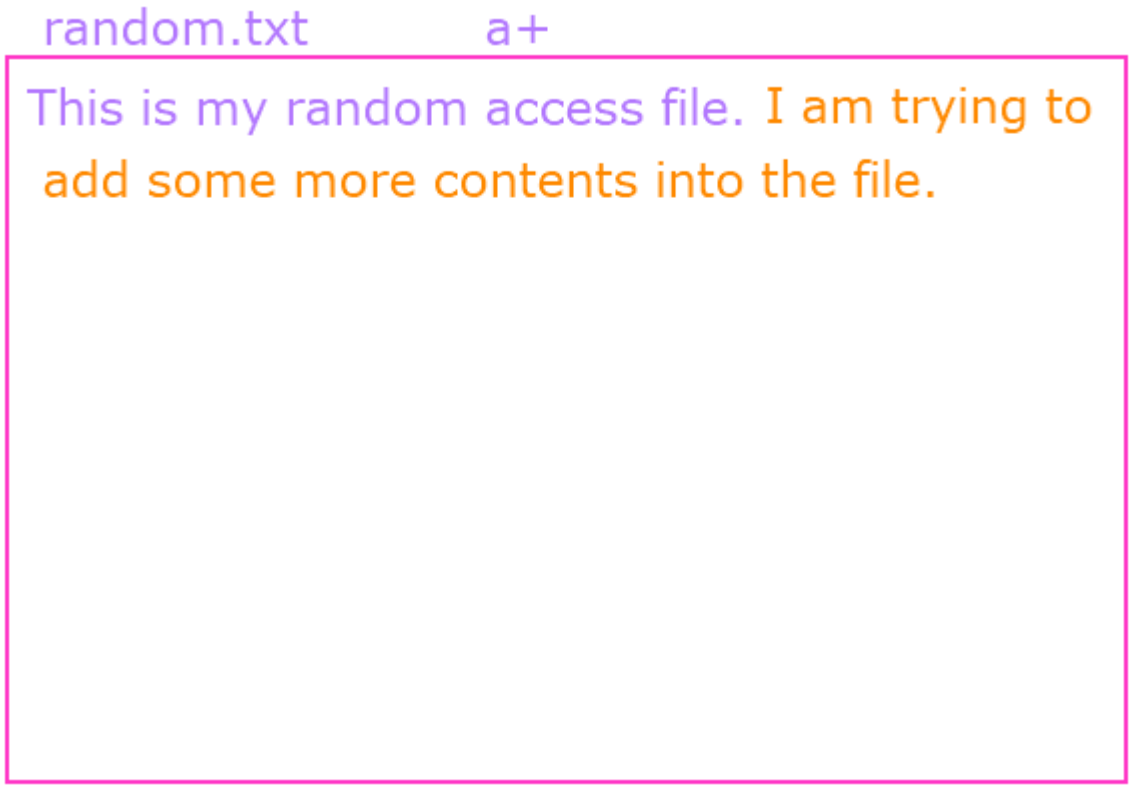
syntax:
void rewind(FILE *filepointer);

Ex:
rewind(fp);

Write a program to demonstrate the Random Access Files.

> Save some text to file.
> Get the position of cursor using ftell()
> Place the cursor at the begining using rewind()
> Get the position of cursor using ftell()
> Place the cursor after 8 characters from the begining of the file
> Get the position of cursor using ftell()
> Read the contents till the end of the file from current location
> Get the position of cursor using ftell()
> Add some more contents to the file at the end
> Get the position of cursor using ftell()

> Place the cursor before 23 characters from end of the file
> Get the position of cursor using ftell()
> Read the contents till the end of the file

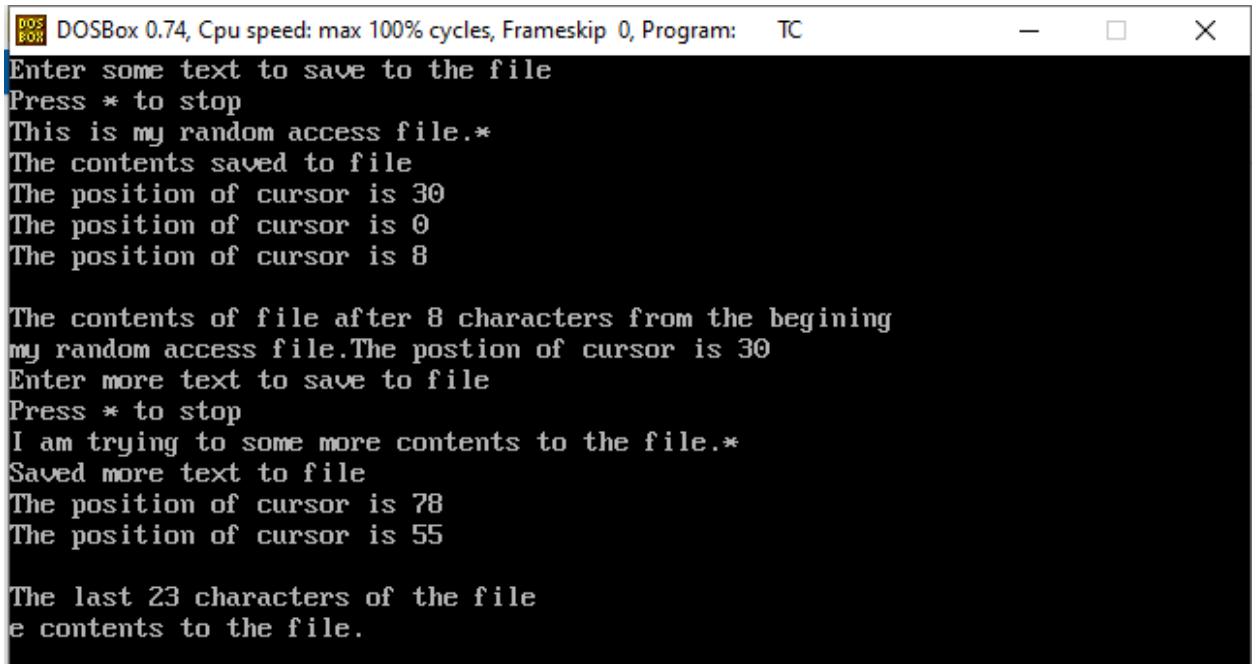


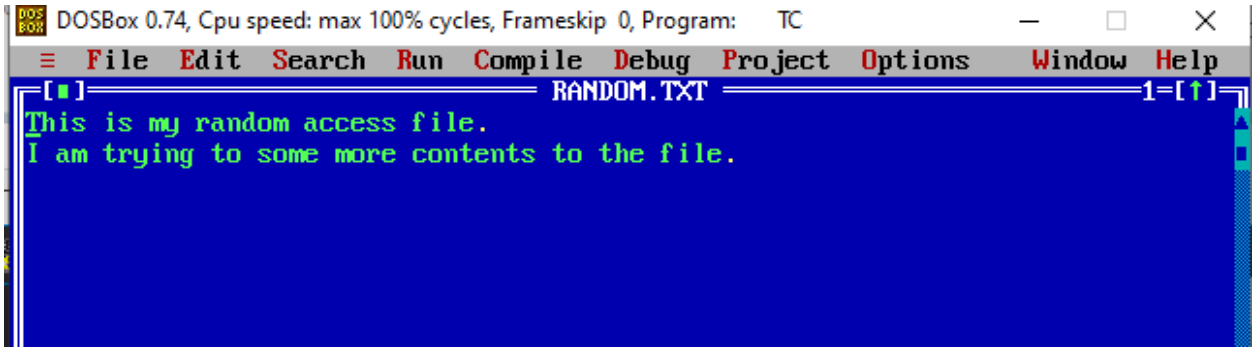
file11.c

```
#include<stdio.h>
main()
{
        FILE *fp;
        char ch;
        long int pos;
        int n;
        clrscr();
        fp=fopen("random.txt","a+");
        if(fp==NULL)
        {
                printf("The file is not created\n");
                exit(0);
        }
        printf("Enter some text to save to the file\n");
        printf("Press * to stop\n");
        while((ch=getchar())!='*')
        {
```

```
            putc(ch,fp);
        }
        printf("The contents saved to file\n");
        pos=ftell(fp);
        printf("The position of cursor is %ld\n",pos);
        rewind(fp);
        pos=ftell(fp);
        printf("The position of cursor is %ld\n",pos);
        fseek(fp,8,SEEK_SET);
        pos=ftell(fp);
        printf("The position of cursor is %ld\n\n",pos);
        printf("The contents of file after 8 characters from the begining\n");
        while((ch=getc(fp))!=EOF)
        {
            putchar(ch);
        }
        pos=ftell(fp);
        printf("The postion of cursor is %ld\n",pos);

        printf("Enter more text to save to file\n");
        printf("Press * to stop\n");
        while((ch=getchar())!='*')
        {
            putc(ch,fp);
        }
        printf("Saved more text to file\n");
        pos=ftell(fp);
        printf("The position of cursor is %ld\n",pos);

        fseek(fp,-23,SEEK_END);
        pos=ftell(fp);
        printf("The position of cursor is %ld\n\n",pos);
        printf("The last 23 characters of the file\n");
        while((ch=getc(fp))!=EOF)
        {
            putchar(ch);
        }

        fclose(fp);
        return 0;
}
```

output:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter some text to save to the file
Press * to stop
This is my random access file.*
The contents saved to file
The position of cursor is 30
The position of cursor is 0
The position of cursor is 8
The contents of file after 8 characters from the begining
my random access file.The postion of cursor is 30
Enter more text to save to file
Press * to stop
I am trying to some more contents to the file.*
Saved more text to file
The position of cursor is 78
The position of cursor is 55
The last 23 characters of the file
e contents to the file.


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
File Edit Search Run Compile Debug Project Options Window Help
RANDOM.TXT
This is my random access file.
I am trying to some more contents to the file.