Q1. Write a program to determine the maximum of 3 numbers.

Code:

```
#include<stdio.h>
void main()
{
        int num1,num2,num3;
        printf("Enter first number:");
        scanf("%d",&num1);
        printf("Enter second number:");
        scanf("%d",&num2);
        printf("Enter third number:");
        scanf("%d",&num3);

        if((num1>num2)&&(num1>num3)){
                printf("%d is the maximum of all three numbers",num1);
        }
        else if((num2>num1)&&(num2>num3)){
                printf("%d is the maximum of all three numbers",num2);
        }
        else{
                printf("%d is the maximum of all three numbers",num3);
        }
}
```

Output:

```
Enter first number:10
Enter second number:20
Enter third number:30
30 is the maximum of all three numbers
```

Q2.Write a program to swap the values of two variables

Code:

```
#include<stdio.h>
void main(){
        int num1,num2,var;
        printf("enter first number");
        scanf("%d",&num1);
        printf("enter second number");
        scanf("%d",&num2);
        printf("Before swapping");
        printf("\nNum1: %d",num1);
        printf("\nNum2: %d",num2);
        var=num1;
        num1=num2;
        num2=var;
        printf("\nAfter swapping");
        printf("\nNum1: %d",num1);
        printf("\nNum2: %d",num2);
}
```

Output:

```
enter first number24
enter second number78
Before swapping
Num1: 24
Num2: 78
After swapping
Num1: 78
Num2: 24
```

Q3. Write a program that reads the percentage obtained by the students and determines and prints the class obtained by the students as per the following rules:

Percentage	Class
0-39	Fail
40-59	Second Class
60-79	First Class
80-100	Distinction

Code:

```
#include<stdio.h>

void main()

{

        int per;

        printf("Enter your percentage");

        scanf("%d",&per);

        if(per>=0 && per<=39){

                printf("Class:Fail");

        }

        else if(per>=40 ||per<=59)

        {

                printf("Class:Second class");

        }

        else if(per>=60 && per<=79)

        {

                printf("Class:First class");

        }

        else if(per>=80 && per<=100)

        {   printf("Class:Distinction");

        }

}
```

Output:

```
Enter your percentage58

Class:Second class
```

Q4.Write a program to calculate the area of circle/rectangle/triangle.

C indicate circle,

R indicate rectangle,

T indicate triangle.(use symbolic constant to define the value of pie.)

Code:

```
#include<stdio.h>

#define PI 3.14159

void main(){

        char ch,c;

        float r,area,l,b,ba,h;

        printf("Press c to calculate area of circle\n ");

        printf("Press r to calculate area of rectangle\n");

        printf("Press t to calculate area of triangle\n");

        printf("Enter choice:");

        scanf("%c",&ch);

        switch(ch){

                case'C':


                        printf("Enter radius of circle\n");

                        scanf("%f",&r);

                        area=PI*r*r;

                        printf("\nArea of circle:%f",area);

                        break;


                case 'R':

                        printf("Enter length of rectangle\n");

                        scanf("%f",&l);

                        printf("Enter width of rectangle\n");

                        scanf("%f",&b);
```

```
                    area=l*b;

                              printf("Area of rectangle:%f",area);

                    break;


          case 'T':

                    printf("Enter base of triangle\n");

                    scanf("%f",&ba);

                    printf("Enter height of triangle\n");

                    scanf("%f",&h);

                    area=1/2*ba*h;





     }

}
```

Output:

```
Press c to calculate area of circle

 Press r to calculate area of rectangle

Press t to calculate area of triangle

Enter choice:C

Enter radius of circle

4


Area of circle:50.265442
```

Q5. Write a program that accept basic,HRA and DA from the user and calculate total salary.

Code:

```
#include <stdio.h>
void main(){
   float bs,da_per,hra_per,gs,da,hra;
   printf("Enter basic salary:");
   scanf("%f",&bs);
   printf("Enter DA:");
   scanf("%f",&da_per);
   printf("Enter HRA:");
   scanf("%f",&hra_per);

   hra=(hra_per/100)*bs;
   da=(da_per/100)*bs;

   gs=da+hra+bs;
   printf("Gross salary is %.2f",gs);
}
```

Output:

```
Enter basic salary:15000
Enter DA:5
Enter HRA:10
Gross salary is 17250.00
```

Q6 Write a program to print multiplication table of a given number

Code:

```
#include<stdio.h>
void main(){
   int n,i,res;
   printf("Enter the number to print its multiplication table\n");
   scanf("%d",&n);
   for(i=1;i<=10;i++){
       res=n*i;
       printf("%d * %d = %d\n",n,i,res);
   }
}
```

Output:

```
Enter the number to print its multiplication table
5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Q7 Write a program to determine given number is prime or not

Code:

```
#include<stdio.h>
void main(){
   int i,n,f=0;
   printf("Enter a number to check whether it is prime or not:");
   scanf("%d",&n);
   for(i=2;i<n;i++){
      if(n % i==0){
         f=1;
         break;
      }
   }
   if(f==1){
      printf("It is not prime number");
   }
   else{
      printf("It is a prime number");
   }
}
```

Output:

```
Enter a number to check whether it is prime or not:29
It is a prime number
```

Q8 Write a program to reverse a given number and display the sum of all digits

Code:

```
#include<stdio.h>
void main(){
        int n,mod,rev=0,sum=0;
        printf("Enter a number");
        scanf("%d",&n);
        while(n>0){
                mod=n%10;
                rev=rev*10+mod;
                sum+=mod;
                n=n/10;
        }
        printf("\nReverse number is:%d",rev);
        printf("\nSum of all digits is:%d",sum);
}
```

Output:

```
Enter a number789

Reverse number is:987
Sum of all digits is:24
```

Q9 Write a program to accept two numbers and perform basic operation of calculater(+,-,*,/).(Use switch case)

Code:

```
#include<stdio.h>

void main(){

        int num1,num2,res;

        char op;

        printf("Enter first number");

        scanf("%d",&num1);

        printf("\nEnter second number");

        scanf("%d",&num2);

        printf("\nEnter + to perform addition");

        printf("\nEnter - to perform subtraction");

        printf("\nEnter * to perform multiplication");

        printf("\nEnter / to perform division");

        printf("\nEnter operator:");

        scanf(" %c",&op);

        switch(op){

                case '+':

                        res=num1+num2;

                        printf("Sum is %d",res);

                        break;


                case '-':

                        res=num1-num2;

                        printf("Result is %d",res);

                        break;


                case '*':

                        res=num1*num2;

                        printf("Result is %d",res);
                        break;


                case '/':

                        res=num1+num2;

                        printf("Result is %d",res);

                        break;

        }

}
```

Output:

```
Enter first number15

Enter second number30

Enter + to perform addition

Enter - to perform subtraction

Enter * to perform multiplication

Enter / to perform division

Enter operator:*

Result is 450
```

Q10 Write a program to find maximum and minimum element from 1-Dimensional array

Code:

```
#include<stdio.h>

void main(){

   int i, min ,max,n;

   int a[10];

    printf("Enter length of array");

   scanf("%d",&n);

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

       printf("\nEnter element at[%d] position",i);

       scanf("%d",&a[i]);    }

  min=max=a[0];

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

       if(a[i]>max){

           max=a[i];       }

       if(a[i]<min){

           min=a[i];

       }

   }

  printf("\nMaximum:%d",max);

   printf("\nMinimum:%d",min);

}
```

Output:

```
Enter length of array5

Enter element at[0] position24

Enter element at[1] position54

Enter element at[2] position63

Enter element at[3] position14

Enter element at[4] position7

Maximum:63

Minimum:
```

Q11 Write a program to sort given array in ascending order.

Code:

```
#include <stdio.h>

void main() {

   int n,i,j,temp;

   int arr[100];

   printf("Enter the number of elements in the array: ");

   scanf("%d", &n);

   printf("Enter %d elements:\n", n);

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

      scanf("%d", &arr[i]); }

     for (i = 0; i < n - 1; i++) {

      for ( j = 0; j < n - i - 1; j++) {

         if (arr[j] > arr[j + 1]) {

           temp = arr[j];

            arr[j] = arr[j + 1];

            arr[j + 1] = temp; }

      }

   }

 printf("Sorted array in ascending order:\n");

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

      printf("%d ", arr[i]); }

   printf("\n");

}
```

Output:

```
Enter the number of elements in the array: 5

Enter 5 elements:

78

45
```

Q12 Write a program to add two matrices

Code:

```
#include<stdio.h>

void main(){

        int i,j,row,col;

        int a[10][10],b[10][10],c[50][50];

        printf("Enter no of rows for matrices ");

        scanf("%d",&row);

        printf("\nEnter no of cols for matrices ");

        scanf("%d",&col);

        for(i=0;i<row;i++){

                for(j=0;j<col;j++){

                        printf("\nEnter element in first matrix at [%d][%d]:",i,j);

                        scanf("%d",&a[i][j]);   }

        }

        for(i=0;i<row;i++){

                for(j=0;j<col;j++){

                        printf("\nEnter element in second matrix at [%d][%d]:",i,j);

                        scanf("%d",&b[i][j]);}

        }

        for(i=0;i<row;i++)

        {

                for(j=0;j<row;j++)

                {  c[i][j]=a[i][j]+b[i][j];   }

        }

        printf("\nFirst Matrix is :\n");

        for(i=0;i<row;i++){

                for(j=0;j<col;j++){

                        printf("%d",a[i][j]);

                        if(j<col-1)printf(",");}

                printf("\n");
```

```
	}
	printf("\nSecond Matrix is :\n");
	for(i=0;i<row;i++){
		for(j=0;j<col;j++){


			printf("%d",b[i][j]);
			if(j<col-1)printf(",");


		}
		printf("\n");
	}
	printf("\nNew Matrix is :\n");
	for(i=0;i<row;i++){
		for(j=0;j<col;j++){
			printf("%d",c[i][j]);
			if(j<col-1)printf(",");
		}
		printf("\n");
	}
}
```

Output:

```
Enter no of rows for matrices 2
Enter no of cols for matrices 2
Enter element in first matrix at [0][0]:10
Enter element in first matrix at [0][1]:20
Enter element in first matrix at [1][0]:30
Enter element in first matrix at [1][1]:40
Enter element in second matrix at [0][0]:50
Enter element in second matrix at [0][1]:60
Enter element in second matrix at [1][0]:70
Enter element in second matrix at [1][1]:80



First Matrix is :

10,20

30,40

Second Matrix is :

50,60

70,80

New Matrix is :

60,80

100,120
```

Q13 Write a program to find element at given position from 2-Dimensional array.

Code:

```
#include<stdio.h>

void main(){

        int i,j,a[20][20];

        int rpos,cpos,r,c,ele;


        printf("Enter row of array");

        scanf("%d",&c);


        printf("\nEnter column of array");

        scanf("%d",&r);


        for(i=0;i<r;i++)

        {

                for(j=0;j<c;j++)

                {

                        printf("\nEnter element at [%d][%d]",i,j);

                        scanf("%d",&a[i][j]);

                }

        }

        printf("\nEnter the row position ");

        scanf("%d",&rpos);

        printf("\n Enter the column position");

        scanf("%d",&cpos);

        if(rpos<0 || rpos>=r || cpos<0 || cpos>=c){

                printf("Invalid index");


        }

        else{

        ele=a[rpos][cpos];
        printf("\n Element at [%d][%d] position is:%d",rpos,cpos,ele);

        }


}
```

Output:

```
Enter row of array2


Enter column of array2

Enter element at [0][0]12

Enter element at [0][1]36

Enter element at [1][0]78

Enter element at [1][1]93

Enter the row position 1

 Enter the column position1

 Element at [1][1] position is:93
```

Q14 Write a program that will read a text and count all occurrences of a particular character using function.

Code:

```
#include <stdio.h>

int occur(char text[], char ch) {

    int count = 0;

    int i;

    for (i = 0; text[i] != '\0'; i++) {

        if (text[i] == ch) {

            count++;

        }

    }

    return count;

}

void main() {

    char str[100], ch;

    int count;

    printf("Enter a text: ");

    gets(str);

    printf("Enter the character to count: ");

    scanf(" %c", &ch);

    count = occur(str, ch);



    printf("The character '%c' occurs %d times.", ch, count);

}
```

Output:

```
Enter a text: PGDCSSSA

Enter the character to count: S

The character 'S' occurs 3 times.
```

Q15 Write a function which returns 1 if the given number is palindrome otherwise returns 0.

Code:

```
#include <stdio.h>

int isPalindrome(int num) {

   int org = num;

   int rev = 0,mod;

   while (num > 0) {

      mod = num % 10;

      rev = rev * 10 + mod;

      num /= 10;

   }

   return org == rev ? 1 : 0;

}


int main() {

   int num;

   printf("Enter a number: ");

   scanf("%d", &num);




   if (isPalindrome(num)) {

      printf("%d is a palindrome.\n", num);

   } else {

      printf("%d is not a palindrome.\n", num);

   }


   return 0;

}
```

Output:

```
Enter a number: 151

151 is a palindrome.
```

Q16 Write a recursive function for finding the factorial of a number.

Code:

```
#include<stdio.h>

int factorial(int num)

{

        if(num==1)

        {

                return num;

        }

        else

        {

                return num*factorial(num-1);

        }

}

void main(){

        int n,res;

        printf("Enter value of n");

        scanf("%d",&n);

        res=factorial(n);

        printf("Factorial of %d is:%d",n,res);

}
```

Output:

```
Enter value of n5

Factorial of 5 is:120
```

Q17 Write a program to perform summation of all elements of array using pointers.

Code:

```
#include<stdio.h>
void main(){
	int n,i,sum=0;
	int arr[100];
	int *ptr=arr;
	printf("Enter no of elements");
	scanf("%d",&n);

	for(i=0;i<n;i++){
		printf("Enter integer");
		scanf("%d",ptr+i);
	}
	for(i=0;i<n;i++){
		sum+=*(ptr+i);

	}
	printf("Sum:%d",sum);
}
```

Output:

```
Enter no of elements5
Enter integer14
Enter integer56
Enter integer3
Enter integer45
Enter integer89
Sum:207
```

Q18 Write a function using pointers to exchange the value stored in two locations in the memory.

Output:

```
#include <stdio.h>

void swap(int *a, int *b) {

   int temp = *a;

   *a = *b;

   *b = temp;

}

void main() {

   int x, y;


   printf("Enter two integers:\n");

   scanf("%d %d", &x, &y);


   printf("Before swapping: x = %d, y = %d\n", x, y);

   swap(&x, &y);


   printf("After swapping: x = %d, y = %d\n", x, y);

}
```

Output:

```
Enter two integers:

12

56

Before swapping: x = 12, y = 56

After swapping: x = 56, y = 12
```

Q19 Write a program to create structure Student with student’s roll no, name and marks of three subjects (Maths, Science and English) and display the details of student with total marks of all subjects along with the percentage and passing class in proper format.

Code:

```
#include<stdio.h>
#include<string.h>
struct Student{
    int rollno;
    char name[15];
    int eng,maths,sci;
    float total,per;
    char passclass[20];
};
void percentage(struct Student* s){
    s->total=s->eng+s->maths+s->sci;
    s->per=s->total/3.0;

    if(s->per>=60){
        strcpy(s->passclass,"First class");
    }
    else if(s->per>=50){
        strcpy(s->passclass,"Second class");
    }
    else if(s->per>=40){
        strcpy(s->passclass,"Third class");
    }

}
void display(struct Student s){
    printf("------Marksheet----------\n");
```

```
	printf("Student rollno:%d\n",s.rollno);
	printf("Student name:%s\n",s.name);
	printf("Maths marks:%d\n",s.maths);
	printf("Maths marks:%d\n",s.sci);
	printf("Maths marks:%d\n",s.eng);
	printf("Maths marks:%f\n",s.per);
	printf("Division:%s\n",s.passclass);
	printf("----------------\n");
}
int main(){
	int n,i;
	printf("Enter the numbe of student:");
	scanf("%d",&n);

	struct Student s[n];

	for(i=0;i<n;i++){
		printf("\nEnter Student rollno:");
		scanf("%d",&s[i].rollno);
		printf("\nEnter Student name:");
		scanf("%s",&s[i].name);
		printf("\nEnter English marks:");
		scanf("%d",&s[i].eng);
		printf("\nEnter Science marks:");
		scanf("%d",&s[i].sci);
		printf("\nEnter Maths marks:");
		scanf("%d",&s[i].maths);
		printf("\n");
		percentage(&s[i]);
	}
	for(i=0;i<n;i++){
		display(s[i]);
	}
	return 0;
}
```

Output:

```
Enter the numbe of student:3
Enter Student rollno:1
Enter Student name:Dev
Enter English marks:78
Enter Science marks:80
Enter Maths marks:73
Enter Student rollno:2
Enter Student name:Stuti
Enter English marks:45
Enter Science marks:79
Enter Maths marks:83
Enter Student rollno:3
Enter Student name:Pari
Enter English marks:67
Enter Science marks:71
Enter Maths marks:82

------Marksheet----------
Student rollno:1
Student name:Dev
Maths marks:73
Maths marks:80
Maths marks:78
Maths marks:77.000000
```

```
Division:First class

----------------

------Marksheet----------

Student rollno:2

Student name:Stuti

Maths marks:83

Maths marks:79

Maths marks:45

Maths marks:69.000000

Division:First class

----------------

------Marksheet----------

Student rollno:3

Student name:Pari

Maths marks:82

Maths marks:71

Maths marks:67

Maths marks:73.333336

Division:First class

----------------
```

Q20 Write a program to create structure Time (data members : int h, int m, int sec). Read a value as seconds from user to display new time after adding seconds to Time structure.

Code:

```
#include<stdio.h>

struct Time{

        int h;

        int m;

        int sec;

};

int main(){

        struct Time t;

        int addsec;

        printf("Enter the current time in hours,minutes and seconds:");

        scanf("%d %d %d",&t.h,&t.m,&t.sec);


        printf("Enter seconds to add to current time");

        scanf("%d",&addsec);


        t.sec += addsec;


        t.m +=t.sec/60;

        t.sec=t.sec%60;

        t.h+=t.m/60;

        t.m=t.m%60;


        printf("Updated time is:- %d : %d : %d",t.h,t.m,t.sec);

        return 0;

}
```

Output:

```
Enter the current time in hours,minutes and seconds:12

5

14

Enter seconds to add to current time564

Updated time is:- 12 : 14 : 38
```

Q21 Write a program to define a structure called book. Write a program to read information about 5 books and display books details in ascending order of price in proper format.

Code:

```
#include<stdio.h>

struct Book{


        char name[50];

        char authorname[50];

        int price;

};

void sortbook(struct Book b[]) {

  struct Book temp;

  int i,j;

  for (i = 0; i < 5 - 1; i++) {

    for (j = 0; j < 5 - i - 1; j++) {

      if (b[j].price > b[j + 1].price) {

        temp = b[j];

        b[j] = b[j + 1];

        b[j + 1] = temp;

      }

    }

  }

}

int main(){


        int i;

        struct Book b[5];

        for(i=0;i<5;i++){

                printf("Book No:%d\n",i+1);

                printf("Enter Book name:");
                scanf("%s",&b[i].name);
                printf("Enter Author name:");
                scanf("%s",&b[i].authorname);
                printf("Enter price");
                scanf("%d",&b[i].price);
        }
        printf("\n");

        sortbook(b);
        for(i=0;i<5;i++){
                printf("\n--------Book Details-----------");
                printf("\nBook name:%s",b[i].name);
                printf("\nAuthor name:%s",b[i].authorname);
                printf("\nPrice:%d",b[i].price);
                printf("\n--------------------------------");

        }

        return 0;

}
```

Output:

```
Book No:1
Enter Book name:maths
Enter Author name:abc
Enter price500
Book No:2
Enter Book name:science
Enter Author name:xyz
Enter price450
Book No:3
```

```
Enter Book name:english

Enter Author name:pqr

Enter price300

Book No:4

Enter Book name:economics

Enter Author name:mno

Enter price200

Book No:5

Enter Book name:chemistry

Enter Author name:abc

Enter price340


--------Book Details-----------

Book name:economics

Author name:mno

Price:200

--------------------------------

--------Book Details-----------

Book name:english

Author name:pqr

Price:300

--------------------------------

--------Book Details-----------

Book name:chemistry

Author name:abc

Price:340

--------------------------------

--------Book Details-----------

Book name:science

Author name:xyz

Price:450
```

```
--------------------------------

--------Book Details-----------

Book name:maths

Author name:abc

Price:500

--------------------------------
```

Q22 Write a program to copy contents of one file to another and also print the no of lines in the first file.

Code:

```
#include<stdio.h>
int main(){
        FILE *f1,*f2;
        char ch;
        int count=1;

        f1=fopen("abc.txt","r");
        if(f1==NULL){
                printf("Error opening file1");
                return 1;
        }

        f2=fopen("xyz.txt","w");
        if(f2==NULL){
                printf("Error opening file2");
                return 1;
        }

        while((ch=fgetc(f1))!= EOF){
                fputc(ch,f2);
                if(ch=='\n'){
                        count++;
                }
        }

        fclose(f1);
        fclose(f2);
        printf("No of lines in the source file: %d\n",count);

        printf("Content copied successfully");

        return 0;

}
```

abc.txt

A photograph (also known as a photo, image, or picture) is an image created by light falling on a photosensitive surface, usually photographic film or an electronic image sensor, such as a CCD or a CMOS chip. Most photographs are now created using a smartphone or camera, which uses a lens to focus the scene's visible wavelengths of light into a reproduction of what the human eye would see.

The process and practice of creating such images is called photography.

The first permanent photograph, a contact-exposed copy of an engraving, was made in 1822 using the bitumen-based "heliography" process developed by Nicéphore Niépce

Xyz.txt

A photograph (also known as a photo, image, or picture) is an image created by light falling on a photosensitive surface, usually photographic film or an electronic image sensor, such as a CCD or a CMOS chip. Most photographs are now created using a smartphone or camera, which uses a lens to focus the scene's visible wavelengths of light into a reproduction of what the human eye would see.

The process and practice of creating such images is called photography.

The first permanent photograph, a contact-exposed copy of an engraving, was made in 1822 using the bitumen-based "heliography" process developed by Nicéphore Niépce

Output:

```
No of lines in the source file: 3

Content copied successfully
```

Q 23 Write a function to read a file and count the no of characters , spaces, newlines and no .of words in a given text file.

Code:

```
#include <stdio.h>

void countFile(const char *filename) {

    FILE *file = fopen(filename, "r");

    char ch;

    int characters = 0, spaces = 0, newlines = 1, words = 0;

    int inWord = 0;



    if (file == NULL) {

        printf("Error opening file.\n");

        return;

    }



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

        characters++;



        if (ch == ' ') {

            spaces++;

        } else if (ch == '\n') {

            newlines++;

        }

        if (ch == ' ' || ch == '\n' || ch == '\t') {

            inWord = 0;

        } else {

            if (inWord == 0) {

                inWord = 1;

                words++;

            }
        }

    }


    fclose(file);

    printf("Number of characters: %d\n", characters);

    printf("Number of spaces: %d\n", spaces);

    printf("Number of newlines: %d\n", newlines);

    printf("Number of words: %d\n", words);

}


int main() {

    const char *filename = "abc.txt";

    countFile(filename);

    return 0;

}
```

Output:

```
Number of characters: 632

Number of spaces: 101

Number of newlines: 3

Number of words: 103
```

Q24 Write an interactive meu driven program that will access the data file created in the above problem to do one of the following tasks:

a.Determine the telephone number of a specific customers

b. Determine the customer whose telephone no is specified

c.Add a new record

d.Delete a record

e.Generate the listing of all customers and their telephone numbers

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct Customer {

   char name[20];

   char telephone[10];

};


struct Customer *custs = NULL;

struct Customer s;

FILE *file;

int n = 0;


void printRecord(struct Customer record) {

   printf("\n--------\n");

   printf("Name: %s\n", record.name);

   printf("Telephone Number: %s\n", record.telephone);

   printf("----------\n");

}


void printall(){
```

```
            int i;

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

                        printRecord(custs[i]);

            }

}


void extractData() {

   while (fread(&s, sizeof(s), 1, file) == 1) {

      struct Customer *temp = realloc(custs, (n + 1) * sizeof(struct Customer));

      if (temp == NULL) {

         printf("Memory allocation failed!\n");

         exit(1);

      }

      custs = temp;

      custs[n++] = s;

   }

   printf("\nData extracted from the file.\n");

}


void openFile() {

   char filename[50];

   printf("Enter the name of the file to open or create: ");

   fgets(filename, sizeof(filename), stdin);

   filename[strcspn(filename, "\n")] = 0;


   file = fopen(filename, "a+b");

   if (!file) {

      printf("Could not open file!\n");

      return;

   }
   printf("File opened successfully.\n");

   extractData();

}




void writeBinary() {

            int i;

   if (file) {

      fseek(file, 0, SEEK_SET);

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

         fwrite(&custs[i], sizeof(struct Customer), 1, file);

      }

      printf("\nSuccessfully saved data to the file.\n");

   } else {

      printf("File is not opened!\n");

   }

}


void inputRecord() {

   printf("\nEnter the details of the new customer:\n");

   printf("Enter the name: ");

   fgets(s.name, sizeof(s.name), stdin);

   s.name[strcspn(s.name, "\n")] = 0;


   printf("Enter the telephone number: ");

   fgets(s.telephone, sizeof(s.telephone), stdin);

   s.telephone[strcspn(s.telephone, "\n")] = 0;


   struct Customer *temp = realloc(custs, (n + 1) * sizeof(struct Customer));

   if (temp == NULL) {

      printf("Memory allocation failed!\n");
```

```
        exit(1);

    }

    custs = temp;

    custs[n++] = s;

    printf("\nRecord added to the list.\n");

}




void deleteRecord() {

            int i, j;

    char str[20];

    printf("Enter the mobile number or the name of the customer to delete: ");

    fgets(str, sizeof(str), stdin);

    str[strcspn(str, "\n")] = 0;


    int found = 0;

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

        if (strcmp(custs[i].name, str) == 0 || strcmp(custs[i].telephone, str) == 0) {

            for (j = i; j < n - 1; j++) {

                custs[j] = custs[j + 1];

            }

            n--;

            struct Customer *temp = realloc(custs, n * sizeof(struct Customer));

            if (temp == NULL && n > 0) {

                printf("Memory allocation failed!\n");

                exit(1);

            }

            custs = temp;

            found = 1;

            printf("Record deleted.\n");

            break;
        }
    }
    if (!found) {
        printf("No such record found to delete.\n");
    }
}
void getRecord() {
            int i;
    char str[20];
    printf("\nEnter the telephone number or name: ");
    fgets(str, sizeof(str), stdin);
    str[strcspn(str, "\n")] = 0;

    for (i = 0; i < n; i++) {
        if (strcmp(custs[i].name, str) == 0 || strcmp(custs[i].telephone, str) == 0) {
            printRecord(custs[i]);
            getchar();
            return;
        }
    }
    printf("No such record found.\n");
    getchar();
}



void menu() {
            if(!file) openFile();
    char c;
    printf("\n--- MENU ---");
    printf("\nOpen or create file (o)\n");
    printf("Add new record (a)\n");
    printf("Get record (g)\n");

    printf("Delete customer (d)\n");

    printf("Save changes (s)\n");

    printf("Print all record (l)\n");

    printf("Exit (x)\n");

    printf("Input: ");

    scanf(" %c", &c);

    while (getchar() != '\n');


    switch (c) {

        case 'a': inputRecord(); break;

        case 'g': getRecord(); break;

        case 'd': deleteRecord(); break;

        case 's': writeBinary(); break;

        case 'o': openFile(); break;

        case 'l': printall(); break;

        case 'x':

            if (file) fclose(file);

            free(custs);

            exit(0);

        default: printf("\nInvalid input!!!\n");

    }

    menu();

}



int main() {

    menu();

    free(custs);

    return 0;

}
```

Output:

```
Enter the name of the file to open or create: customer.bin
File opened successfully.

Data extracted from the file.

--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input: l

--------
Name: Raj

Telephone Number: 1234567890
----------

--------
Name: Priya

Telephone Number: 9087654321
----------

--- MENU ---
Open or create file (o)
```

```
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input: a

Enter the details of the new customer:
Enter the name: Om
Enter the telephone number: 1243568790

Record added to the list.

--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input:
Invalid input!!!

--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
```

```
Print all record (l)

Exit (x)

Input: g


Enter the telephone number or name: Om


--------

Name: Om

Telephone Number: 124356879

----------




--- MENU ---

Open or create file (o)

Add new record (a)

Get record (g)

Delete customer (d)

Save changes (s)

Print all record (l)

Exit (x)

Input: d

Enter the mobile number or the name of the customer to delete: Raj

No such record found to delete.


--- MENU ---

Open or create file (o)

Add new record (a)

Get record (g)

Delete customer (d)

Save changes (s)

Print all record (l)
```

```
Exit (x)
Input: d
Enter the mobile number or the name of the customer to delete: Om
Record deleted.

--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input: s

Successfully saved data to the file.

--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input: l

--------
Name: Raj

Telephone Number: 1234567890
```

```
----------


--------
Name: Priya


Telephone Number: 9087654321
----------


--- MENU ---
Open or create file (o)
Add new record (a)
Get record (g)
Delete customer (d)
Save changes (s)
Print all record (l)
Exit (x)
Input: x
```

Q25 Use a structure of Employee to write records of employee to a file. Include a menu that will allow the user to select any of the following feature.

a.Add a new record

b.Delete a record

c. Modify an existing record

d. Retrive and display an entire record for a given Id/Name

e.Generate a complete list of all employees names,addresses and telephone numbers.

f.End of computation/Exit

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


#define MAX_EMPLOYEES 100

#define FILENAME "employees.dat"


struct Employee {

   int id;

   char name[50];

   char address[100];

   char phone[15];

};


void addRecord();

void deleteRecord();

void modifyRecord();

void retrieveRecord();

void listRecords();

void clearBuffer();
```

```
int main() {
   int choice;

   do {
      printf("\nEmployee Management System\n");
      printf("1. Add a new record\n");
      printf("2. Delete a record\n");
      printf("3. Modify an existing record\n");
      printf("4. Retrieve and display a record\n");
      printf("5. Generate a complete list of employees\n");
      printf("6. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
      clearBuffer();

      switch (choice) {
         case 1: addRecord(); break;
         case 2: deleteRecord(); break;
         case 3: modifyRecord(); break;
         case 4: retrieveRecord(); break;
         case 5: listRecords(); break;
         case 6: printf("Exiting...\n"); break;
         default: printf("Invalid choice! Please try again.\n");
      }
   } while (choice != 6);

   return 0;
}

void clearBuffer() {
   while (getchar() != '\n');

}


void addRecord() {

   struct Employee emp;

   FILE *file = fopen(FILENAME, "ab");


   if (file == NULL) {

      printf("Error opening file!\n");

      return;

   }


   printf("Enter ID: ");

   scanf("%d", &emp.id);

   clearBuffer();

   printf("Enter Name: ");

   fgets(emp.name, sizeof(emp.name), stdin);

   strtok(emp.name, "\n");

   printf("Enter Address: ");

   fgets(emp.address, sizeof(emp.address), stdin);

   strtok(emp.address, "\n");

   printf("Enter Phone: ");

   fgets(emp.phone, sizeof(emp.phone), stdin);

   strtok(emp.phone, "\n");


   fwrite(&emp, sizeof(struct Employee), 1, file);

   fclose(file);

   printf("Record added successfully.\n");

}


void deleteRecord() {
   int id;

   struct Employee emp;

   FILE *file = fopen(FILENAME, "rb");

   FILE *tempFile = fopen("temp.dat", "wb");


   if (file == NULL || tempFile == NULL) {

      printf("Error opening file!\n");

      return;

   }


   printf("Enter ID of the employee to delete: ");

   scanf("%d", &id);


   int found = 0;

   while (fread(&emp, sizeof(struct Employee), 1, file)) {

      if (emp.id != id) {

          fwrite(&emp, sizeof(struct Employee), 1, tempFile);

      } else {

          found = 1;

      }

   }


   fclose(file);

   fclose(tempFile);

   remove(FILENAME);

   rename("temp.dat", FILENAME);


   if (found) {

      printf("Record deleted successfully.\n");

   } else {

      printf("Record with ID %d not found.\n", id);
```

```
  }

}


void modifyRecord() {

  int id;

  struct Employee emp;

  FILE *file = fopen(FILENAME, "r+b");


  if (file == NULL) {

    printf("Error opening file!\n");

    return;

  }


  printf("Enter ID of the employee to modify: ");

  scanf("%d", &id);


  int found = 0;

  while (fread(&emp, sizeof(struct Employee), 1, file)) {

    if (emp.id == id) {

      found = 1;

      printf("Enter new Name: ");

      clearBuffer();

      fgets(emp.name, sizeof(emp.name), stdin);

      strtok(emp.name, "\n"); // Remove newline character

      printf("Enter new Address: ");

      fgets(emp.address, sizeof(emp.address), stdin);

      strtok(emp.address, "\n"); // Remove newline character

      printf("Enter new Phone: ");

      fgets(emp.phone, sizeof(emp.phone), stdin);

      strtok(emp.phone, "\n"); // Remove newline character
```

```
            fseek(file, -sizeof(struct Employee), SEEK_CUR); // Move to the record's position
            fwrite(&emp, sizeof(struct Employee), 1, file);
            break;
        }
    }

    fclose(file);

    if (found) {
        printf("Record modified successfully.\n");
    } else {
        printf("Record with ID %d not found.\n", id);
    }
}

void retrieveRecord() {
    int id;
    struct Employee emp;
    FILE *file = fopen(FILENAME, "rb");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter ID of the employee to retrieve: ");
    scanf("%d", &id);

    int found = 0;
    while (fread(&emp, sizeof(struct Employee), 1, file)) {
        if (emp.id == id) {
            found = 1;

            printf("ID: %d\n", emp.id);

            printf("Name: %s\n", emp.name);

            printf("Address: %s\n", emp.address);

            printf("Phone: %s\n", emp.phone);

            break;

        }

    }


    fclose(file);


    if (!found) {

        printf("Record with ID %d not found.\n", id);

    }

}


void listRecords() {

    struct Employee emp;

    FILE *file = fopen(FILENAME, "rb");


    if (file == NULL) {

        printf("Error opening file!\n");

        return;

    }


    printf("\nList of Employees:\n");

    while (fread(&emp, sizeof(struct Employee), 1, file)) {

        printf("ID: %d, Name: %s, Address: %s, Phone: %s\n", emp.id, emp.name, emp.address, emp.phone);

    }


    fclose(file);
```

```
}
```

Output:

```
Employee Management System
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display a record
5. Generate a complete list of employees
6. Exit
Enter your choice: 1
Enter ID: 1
Enter Name: Dev
Enter Address: Pune
Enter Phone: 7896543218
Record added successfully.

Employee Management System
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display a record
5. Generate a complete list of employees
6. Exit
Enter your choice: 4
Enter ID of the employee to retrieve: 1
ID: 1
Name: Dev
Address: Pune
Phone: 7896543218
```

```
Employee Management System

1. Add a new record

2. Delete a record

3. Modify an existing record

4. Retrieve and display a record

5. Generate a complete list of employees

6. Exit

Enter your choice: 1

Enter ID: 2

Enter Name: Disha

Enter Address: Goa

Enter Phone: 2345678901

Record added successfully.


Employee Management System

1. Add a new record

2. Delete a record

3. Modify an existing record

4. Retrieve and display a record

5. Generate a complete list of employees

6. Exit

Enter your choice: 3

Enter ID of the employee to modify: 2

Enter new Name: Disha

Enter new Address: Gandhinagar

Enter new Phone: 2345678901

Record modified successfully.


Employee Management System

1. Add a new record

2. Delete a record
```

```
3. Modify an existing record
4. Retrieve and display a record
5. Generate a complete list of employees
6. Exit
Enter your choice: 5

List of Employees:
ID: 1, Name: Dev, Address: Pune, Phone: 7896543218
ID: 2, Name: Disha, Address: Gandhinagar, Phone: 2345678901

Employee Management System
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display a record
5. Generate a complete list of employees
6. Exit
Enter your choice: 4
Enter ID of the employee to retrieve: 2
ID: 2
Name: Disha
Address: Gandhinagar
Phone: 2345678901

Employee Management System
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display a record
5. Generate a complete list of employees
6. Exit
```

```
Enter your choice: 2

Enter ID of the employee to delete: 2

Record deleted successfully.


Employee Management System

1. Add a new record

2. Delete a record

3. Modify an existing record

4. Retrieve and display a record

5. Generate a complete list of employees

6. Exit

Enter your choice: 6

Exiting...
```