Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

I/P:
```
#include <stdio.h>
int main() {
   int num1, num2, num3, max;


   printf("Enter three numbers: ");
   scanf("%d %d %d", &num1, &num2, &num3);


   if (num1 > num2 && num1 > num3)
      max = num1;
   else if (num2 > num3)
      max = num2;
   else
      max = num3;


   printf("The maximum number is: %d\n", max);

   return 0;
}
```
O/P:
```
Enter three numbers: 5
6
1
The maximum number is: 6
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

I/P:

```
#include<stdio.h>
void main()
{
        int a,b,c;
        printf("Enter two numbers: ");
  scanf("%d %d", &a, &b);
        c=a;
        a=b;
        b=c;
        printf("Value of A is %d",a);
        printf("Value of B is %d",b);


}
```

Output:

```
Enter two numbers: 3
4
Value of A is 4Value of B is 3
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

Percentage Class

0 - 39 Fail

40 - 59 Second class

60 - 79 First class

80 - 100 Distinction

```
#include <stdio.h>

int main() {
    float percentage;


    printf("Enter the percentage obtained by the student: ");
    scanf("%f", &percentage);


    if (percentage >= 0 && percentage < 40) {
        printf("Class: Fail\n");
    } else if (percentage >= 40 && percentage < 60) {
        printf("Class: Second class\n");
    } else if (percentage >= 60 && percentage < 80) {
        printf("Class: First class\n");
    } else if (percentage >= 80 && percentage <= 100) {
        printf("Class: Distinction\n");
    } else {
        printf("Invalid percentage! Please enter a value between 0 and 100.\n");
    }

    return 0;
}
```

Output:

```
Enter the percentage obtained by the student: 89
Class: Distinction
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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.

I/P:

```
#include<stdio.h>


int main() {
   const float p = 3.14;
   float r, b, h;



   printf("\n Enter radius: ");
   scanf("%f", &r);



   printf("\n Enter Base and height: ");
   scanf("%f %f", &b, &h);



   float res;
   res = p * r * r;
   printf("\n Area of the Circle is: %f", res);
```

```
    res = (1.0 / 2) * b * h;
    printf("\n Area of the Triangle is: %f", res);

    return 0;
}
```

```
Output: Enter radius: 2

 Enter Base and height: 1 1

 Area of the Circle is: 12.560000
 Area of the Triangle is: 0.500000
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

```
#include<stdio.h>
void main()
{
        float bs,da,hra,gs;
        printf("Enter your basic salary");
        scanf("%f",&bs);
        printf("Enter your Da");
        scanf("%f",&da);
        printf("Enter your Hra");
        scanf("%f",&hra);
        if(bs<1500)
        {
                da=bs*10/100;
                hra=bs*90/100;
        }
        else
        {
                da=500;
                hra=bs*98/100;
        }
        gs=da+hra+bs;
        printf("Gross salary is %f",gs);
}
```

```
Enter your basic salary15000
Enter your Da2000
Enter your Hra2000

Gross salary is 30200.000000
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

Input:
```
#include<stdio.h>
void main()
{
        int i,n,res;
        printf("Enter number to print multiplication table of that no");
        scanf("%d",&n);
        for(i=1;i<=10;i++)
        {
                res=n*i;
                printf(" \n  %d * %d = %d",n,i,res);
        }

```
}**Output**: Enter number to print multiplication table of that no2

```
2 * 1 = 2
 2 * 2 = 4
 2 * 3 = 6
 2 * 4 = 8
 2 * 5 = 10
 2 * 6 = 12
 2 * 7 = 14
 2 * 8 = 16
 2 * 9 = 18
 2 * 10 = 20
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

Input:
```
#include<stdio.h>
void main()
{
        int n=7,flag=1,i;
        printf("Enter a no to determine no is prime or not");
        scanf("%d",&n);
        for(i=2;i<n;i++)
        {
                if(n%i==0)
                {
                flag=0;
                break;
                }


        }
        if(flag==0)
        printf("%d not prime",n);
        else
        printf("%d is prime",n);
}
```

Output:
```
Enter a no to determine no is prime or not5
5 is prime
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

Input:
```
#include<stdio.h>
void main()
{
        int n,mod,sum=0;
        printf("Enter a number and do reverse of it ");
        scanf("%d",&n);
        printf("Reverse no is ");
        while(n>0)
        {
                mod=n%10;
                sum=sum+mod;
                printf("%d",mod);
                n=n/10;


        }
        printf("\n Sum is %d",sum);
}
```

Output:
```
Enter a number and do reverse of it 123
Reverse no is 321
 Sum is 6
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 1:

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

Input:

```
#include <stdio.h>
int main() {
   float num1, num2, result;
   char operation;



   printf("Enter two numbers: ");
   scanf("%f %f", &num1, &num2);



   printf("Enter operation (+, -, *, /): ");
   scanf(" %c", &operation);



   switch (operation) {
      case '+':
         result = num1 + num2;
         printf("Result: %.2f\n", result);
         break;

      case '-':
         result = num1 - num2;
         printf("Result: %.2f\n", result);
```

```
            break;

        case '*':
            result = num1 * num2;
            printf("Result: %.2f\n", result);
            break;

        case '/':
            if (num2 != 0) {
                result = num1 / num2;
                printf("Result: %.2f\n", result);
            } else {
                printf("Error: Division by zero is not allowed.\n");
            }
            break;

        default:
            printf("Invalid operation!\n");
            break;
    }

    return 0;
}
```

Output:

```
Enter two numbers: 5
3
Enter operation (+, -, *, /): +
Result: 8.00
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

Q10 Write a program to find maximum and minimum element from

1-Dimensional array.

I/P:

```
 #include<stdio.h>
//Maximum Element
void main()
{
	int i,a[20],n,max;
	printf("Enter no of elements you want to enter");
	scanf("%d",&n);
	printf("Enter elements");
	for(i=0;i<n;i++)
	{
		scanf("%d",&a[i]);
	}
	for(i=0;i<n;i++)
	{
		printf("\n %d",a[i]);
	}
	max=a[0];
	for(i=0;i<n;i++)
	{
		if(a[i]>max)
		{
			max=a[i];
		}
```

```
        }
        printf("\n Largest element is %d",max);


}
```

```
output:Enter no of elements you want to enter3
Enter elements12
10
11


 12
 10
 11
 Largest element is 12
```

```
#include<stdio.h>


//Minimum Element
void main()
{
        int i,a[20],n,min;
        printf("Enter no of elements you want to enter");
        scanf("%d",&n);
        printf("Enter elements");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        for(i=0;i<n;i++)
        {
                printf("\n %d",a[i]);
```

```
    }
    min=a[0];
    for(i=0;i<n;i++)
    {
        if(a[i]<min)
        {
            min=a[i];
        }
    }
    printf("\n Smallest element is %d",min);

}
```

Output:

```
Enter no of elements you want to enter3
Enter elements12
11
10

 12
 11
 10
 Smallest element is 10
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

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

I/P:#include<stdio.h>

```
void main()
{
    int n,i,j,temp,no,a[20];
    printf("Enter no of elements");
    scanf("%d",&n);
    printf("Enter elements");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
```

```
            for(i=0;i<n;i++)
            {
                        printf(" \n %d",a[i]);
            }
}
```

output:Enter no of elements3

```
Enter elements10
11
8

 8
 10
 11
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

Q12 Write a program to add two matrices.

Input:

```
#include<stdio.h>
void main()
{
	int i,j;
	int rows,cols;
	int a[10] [10];
	int b[10][10];
	int c[10][10];
	printf("Enter no of rows");
	scanf("%d",&rows);
	printf("Enter no of columns");
	scanf("%d",&cols);
	printf("\n input value for a[] []");
	for(i=0;i<rows;i++)
	{
		for(j=0;j<cols;j++)
		{
			printf("Enter element at index [%d] [%d]",i,j);
			scanf("%d",&a[i][j]);
		}

	}
```

```
        printf("\n input value for b[] []");
for(i=0;i<rows;i++)
{
        for(j=0;j<cols;j++)
        {
                printf("Enter element at index [%d] [%d]",i,j);
                scanf("%d",&b[i][j]);
        }



}
printf("\n Print value for a");
for(i=0;i<rows;i++)
{
        for(j=0;j<cols;j++)
        {
                printf(" %d ",a[i][j]);
        }
        printf("\n");
}

printf("\n Print value for b");
for(i=0;i<rows;i++)
{
        for(j=0;j<cols;j++)
        {
                printf(" %d ",b[i][j]);
        }
        printf("\n");
}
```

```
	printf("\n Add two matrix");
	for(i=0;i<rows;i++)
	{
		for(j=0;j<cols;j++)
		{
			c[i][j]=a[i][j]+b[i][j];
		}
	}

	printf("\n Print value for c");
	for(i=0;i<rows;i++)
	{
		for(j=0;j<cols;j++)
		{
			printf(" %d ",c[i][j]);
		}
		printf("\n");
	}
}
```

O/P:

```
Enter no of rows2
Enter no of columns2

 input value for a[] []Enter element at index [0] [0]10
Enter element at index [0] [1]11
Enter element at index [1] [0]12
Enter element at index [1] [1]13

 input value for b[] []Enter element at index [0] [0]10
Enter element at index [0] [1]11
Enter element at index [1] [0]12
Enter element at index [1] [1]13
```

```
Print value for a 10  11

12  13


Print value for b 10  11

12  13


Add two matrix

Print value for c 20  22

24  26
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

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

I/P:#include<stdio.h>

```
void main()
{
        int a[5][5];
        int i,j;
        int r,c;
        for(i=0;i<2;i++)
        {
                for(j=0;j<2;j++)
                {
                        printf("Enter value for [%d][%d] index",i,j);
                        scanf("%d",&a[i][j]);
                }

        }
```

output:Enter no of rows2

```
Enter no of columns2

 input value for a[] []Enter element at index [0] [0]10
Enter element at index [0] [1]11
Enter element at index [1] [0]12
Enter element at index [1] [1]13
```

```
 input value for b[] []Enter element at index [0] [0]10
Enter element at index [0] [1]11
Enter element at index [1] [0]12
Enter element at index [1] [1]13

 Print value for a 10  11
 12  13

 Print value for b 10  11
 12  13

 Add two matrix
 Print value for c 20  22
 24  26
```

```
            printf("Enter row index");
            scanf("%d",&r);
            printf("Enter col index");
            scanf("%d",&c);



            printf("Value is %d",a[r][c]);

}
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

Q14 Write a program that will read a text and count all occurrences

of a particular character using function.

I/P:#include<stdio.h>

```
int characount(char s[],char c);
void main()
{
        char s[20];
        char c;
        int chara;
        printf("Enter String");
        gets(s);
        printf("Enter a character");
        c=getchar();
        chara=characount(s,c);
        printf("%c apperars %d times",c,chara);
}

int characount(char s[],char c)
{
        int i,count=0;
        for(i=0;s[i]!='\0';i++)
        {
                if(s[i]==c)
                {
                count++;
                }
```

```
            }
            return count;


 }
```

```
Output:Enter StringPGDCSAA
Enter a characterA
A apperars 2 times
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

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

I/P:

```
#include<stdio.h>
int palindrome(int no);
void main()
{
        int n;
        printf("Enter number");
        scanf("%d",&n);
        if(palindrome(n))
        {
                printf(" \n %d is palindrome",n);
        }
        else
        {
                printf("\n %d is not paindrome",n);

        }
}

int palindrome(int no)
{
        int orig_no,rev=0,mod;
        orig_no=no;
        while(no>0)
        {
                mod=no%10;

                rev=rev*10+mod;

                no=no/10;

        }

        printf(" \n Reverse no is %d",rev);

        if(orig_no==rev)

        {

                return 1;

        }

        else

        {

                return 0;

        }

}
```

Output: Enter number128

```
 Reverse no is 821

 128 is not palindrome
```

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

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 2:

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

I/P:

```
int fact(int no);
void main()
{
        int no,f;
        printf("Enter no for finding factorail");
        scanf("%d",&no);
        f=fact(no);
        printf("fact of %d is %d",no,f);
}

int fact(int no)
{
        if(no==1)
        {
                return no;
        }
        else
        {
                return no*fact(no-1);
        }
}
```

output:

```
Enter no for finding factorail5
fact of 5 is 120
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

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

Input:

```
#include <stdio.h>
int sumArray(int *arr, int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += *(arr + i);
    }
    return sum;
}

int main() {
    int arr[] = {1, 2, 3, 4, 5}; // Example array
    int size = sizeof(arr) / sizeof(arr[0]);

    int sum = sumArray(arr, size);

    printf("The sum of all elements in the array is: %d\n", sum);

    return 0;
}
```

O/P: The sum of all elements in the array is: 15

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

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

Input:

```
#include<stdio.h>
void swap(int* a,int* b);
void main()
{
        int x=10,y=20;
        swap(&x,&y);
        printf("Value of x is %d and y is %d",x,y);
}

void swap(int* a,int* b)
{
        *a=*a+*b;
        *b=*a-*b;
        *a=*a-*b;
}
```

O/P: Value of x is 20 and y is 10

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

19. 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.

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

struct Student {
    int id;
    char name[50];
    float marks[3];
    float totalMarks;
    float percentage;
    char division[20];
};

void calculateResults(struct Student* student) {
    student->totalMarks = student->marks[0] + student->marks[1] + student->marks[2];

    student->percentage = (student->totalMarks / 300) * 100;

    if (student->percentage >= 60) {
        strcpy(student->division, "First Class");
    } else if (student->percentage >= 50) {
        strcpy(student->division, "Second Class");
    } else if (student->percentage >= 40) {
        strcpy(student->division, "Third Class");
    } else {
        strcpy(student->division, "Fail");
    }
}

void displayMarksheet(struct Student student) {
    printf("\n---- Marksheet ----\n");
    printf("ID: %d\n", student.id);
    printf("Name: %s\n", student.name);
    printf("Marks in Subject 1: %.2f\n", student.marks[0]);
    printf("Marks in Subject 2: %.2f\n", student.marks[1]);
    printf("Marks in Subject 3: %.2f\n", student.marks[2]);
    printf("Total Marks: %.2f / 300\n", student.totalMarks);
    printf("Percentage: %.2f%%\n", student.percentage);
    printf("Division: %s\n", student.division);
    printf("-------------------\n");
}

int main() {
    int n, i;
    struct Student students[15];
    printf("Enter the number of students: ");
    scanf("%d", &n);



    for (i = 0; i < n; i++) {
        printf("\nEnter details for student %d\n", i + 1);
        printf("Enter student ID: ");
        scanf("%d", &students[i].id);

        printf("Enter student name: ");
        scanf(" %[^\n]%*c", students[i].name);

        printf("Enter marks for Subject 1: ");
        scanf("%f", &students[i].marks[0]);

        printf("Enter marks for Subject 2: ");
        scanf("%f", &students[i].marks[1]);

        printf("Enter marks for Subject 3: ");
        scanf("%f", &students[i].marks[2]);

        calculateResults(&students[i]);
    }

    for (i = 0; i < n; i++) {
        displayMarksheet(students[i]);
    }

    return 0;
}
```

O/P: Enter the number of students: 2

```
Enter details for student 1
Enter student ID: 1
Enter student name: Waheguru
Enter marks for Subject 1: 98
Enter marks for Subject 2: 95
Enter marks for Subject 3: 99
Enter details for student 2
Enter student ID: 2
Enter student name: Shukrana
Enter marks for Subject 1: 99
Enter marks for Subject 2: 89
Enter marks for Subject 3: 97

---- Marksheet ----
ID: 1
Name: Waheguru
Marks in Subject 1: 98.00
Marks in Subject 2: 95.00
Marks in Subject 3: 99.00
Total Marks: 292.00 / 300
Percentage: 97.33%
Division: First Class
-------------------

---- Marksheet ----
ID: 2
Name: Shukrana
Marks in Subject 1: 99.00
Marks in Subject 2: 89.00
Marks in Subject 3: 97.00
Total Marks: 285.00 / 300
Percentage: 95.00%
Division: First Class
-------------------
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

20. 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.

Input: #include <stdio.h>

```
struct Time {
   int h;
   int m;
   int sec;
};

void addSeconds(struct Time* t, int seconds) {
   t->sec += seconds;
   t->m += t->sec / 60;
   t->sec = t->sec % 60;
   t->h += t->m / 60;
   t->m = t->m % 60;
   t->h = t->h % 24; // To wrap around after 24 hours
}
```

```
void displayTime(struct Time t) {
    printf("Time: %02d:%02d:%02d\n", t.h, t.m, t.sec);
}

int main() {
    struct Time t;
    int seconds;

    printf("Enter time in format HH MM SS: ");
    scanf("%d %d %d", &t.h, &t.m, &t.sec);

    printf("Enter seconds to add: ");
    scanf("%d", &seconds);

    addSeconds(&t, seconds);

    printf("New ");
    displayTime(t);

    return 0;
}
```

O/P:

```
Enter time in format HH MM SS: 08 30 30
Enter seconds to add: 1000
New Time: 08:47:10
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

21. 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.

Input: #include <stdio.h>

```
struct Book {
   char title[20];
   char author[20];
   float price;
};

void sortbyprice(struct Book b[], int n) {
   struct Book temp;
   int i, j;
   for (i = 0; i < n - 1; i++) {
      for (j = 0; j < n - 1; j++) {
         if (b[j + 1].price < b[j].price) {
            temp = b[j + 1];
            b[j + 1] = b[j];
            b[j] = temp;
         }
      }
   }
}

void display(struct Book b) {
   printf("\nBook Details\n");
   printf("Book Name: %s\n", b.title);
   printf("Book Author: %s\n", b.author);
   printf("Book Price: %.2f\n", b.price);
}

int main() {
   struct Book b[5];
   int i, n;

   printf("Enter the number of books: ");
   scanf("%d", &n);

   for (i = 0; i < n; i++) {
      printf("\nBook No %d\n", i + 1);
      printf("Enter Title of the book: ");
      scanf(" %[^\n]%*c", b[i].title);
      printf("Enter Author of the book: ");
      scanf(" %[^\n]%*c", b[i].author);
      printf("Enter price of the book: ");
      scanf("%f", &b[i].price);
   }

   sortbyprice(b, n);

   for (i = 0; i < n; i++) {
      display(b[i]);
   }

   return 0;
}
```

O/P:

```
Enter the number of books: 2

Book No 1
Enter Title of the book: Health
Enter Author of the book: Harry
Enter price of the book: 2000

Book No 2
Enter Title of the book: Wealth
Enter Author of the book: Riya
Enter price of the book: 1500

Book Details
Book Name: Wealth
Book Author: Riya
Book Price: 1500.00

Book Details
Book Name: Health
Book Author: Harry
Book Price: 2000.00
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

22. Write a program to copy the contents of one file to another

and also print the no. of lines in the first file.

I/P:

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

void copyAndCountLines(const char *sourceFile, const char *destFile) {
    FILE *source, *dest;
    int ch, lineCount = 0;

    // Open the source file in read mode
    source = fopen(sourceFile, "r");
    if (source == NULL) {
        printf("Cannot open source file: %s\n", sourceFile);
        exit(1);
    }

    // Open the destination file in write mode
    dest = fopen(destFile, "w");
    if (dest == NULL) {
        printf("Cannot open destination file: %s\n", destFile);
        fclose(source);
        exit(1);
    }

    // Copy contents from source to destination and count lines
    while ((ch = fgetc(source)) != EOF) {
        fputc(ch, dest);
        if (ch == '\n') {
            lineCount++;
        }
    }

    // Close both files
    fclose(source);
    fclose(dest);

    // Print the number of lines in the source file
    printf("Number of lines in the source file: %d\n", lineCount);
}

int main() {
    char sourceFile[100], destFile[100];

    // Get the source file name from the user
    printf("Enter the source file name: ");
    scanf("%s", sourceFile);

    // Get the destination file name from the user
    printf("Enter the destination file name: ");
    scanf("%s", destFile);

    // Call the function to copy contents and count lines
    copyAndCountLines(sourceFile, destFile);

    return 0;
}
```

```
O/P: Enter the source file name: pqr.txt

Enter the destination file name: dest1.txt

Number of lines in the source file: 16
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

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.

Input:
```
#include <stdio.h>
#include <stdlib.h>

void rc(const char *f) {
   FILE *f1;
   int nlines = 0, nc = 0, nw = 0, ns = 0;
   f1 = fopen(f, "r");
   if (f1 == NULL) {
      printf("Cannot read file: %s\n", f);
      exit(0);
   }

   int ch, prev_ch = EOF;
   while ((ch = getc(f1)) != EOF) {
      nc++; // Count characters
      if (ch == ' ') {
         ns++; // Count spaces
         if (prev_ch != ' ' && prev_ch != '\n' && prev_ch != '\t' && prev_ch != EOF) {
            nw++; // Count words
         }
      } else if (ch == '\n') {
         nlines++; // Count newlines
         if (prev_ch != ' ' && prev_ch != '\n' && prev_ch != '\t' && prev_ch != EOF) {
                nw++; // Count words
            }
        } else if (ch == '\t') {
            ns++; // Count tabs as spaces
            if (prev_ch != ' ' && prev_ch != '\n' && prev_ch != '\t' && prev_ch != EOF) {
                nw++; // Count words
            }
        }
        prev_ch = ch;
    }

    // Account for the last word if the file doesn't end with a space/newline/tab
    if (prev_ch != ' ' && prev_ch != '\n' && prev_ch != '\t' && prev_ch != EOF) {
        nw++;
    }

    fclose(f1);

    printf("\nNo of lines: %d", nlines);
    printf("\nNo of characters: %d", nc);
    printf("\nNo of spaces: %d", ns);
    printf("\nNo of words: %d", nw);
}

int main() {
    char f[60];
    printf("Enter filename: ");
    scanf("%59s", f); // Use scanf for safer input instead of gets
    rc(f);
    return 0;
}
```

```
O/P: Enter filename: wes.txt


No of lines: 16

No of characters: 529

No of spaces: 76

No of words: 88
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

24. Write an interactive menu 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 the customers and their telephone numbers

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

struct cust {
   char name[20];
   char phoneno[15];
};

void searchcustomer(FILE *fp1) {
   struct cust c;
   char name[20];
   printf("Enter customer name: ");
   scanf("%s", name);
```

```
    rewind(fp1); // Reset file pointer to the start
    while (fread(&c, sizeof(c), 1, fp1)) {
        if (strcmp(c.name, name) == 0) {
            printf("Customer telephone is %s\n", c.phoneno);
            return;
        }
    }
    printf("Customer not found\n");
}

void searchtelephone(FILE *fp1) {
    struct cust c;
    char phone[15];
    printf("Enter phone number: ");
    scanf("%s", phone);
    rewind(fp1); // Reset file pointer to the start
    while (fread(&c, sizeof(c), 1, fp1)) {
        if (strcmp(c.phoneno, phone) == 0) {
            printf("Customer Name is %s\n", c.name);
            return;
        }
    }
    printf("Phone number not found\n");
}

void addrecord(FILE *fp1) {
    struct cust c;
    printf("\nEnter customer name: ");
    scanf("%s", c.name);
    printf("\nEnter customer phone number: ");
    scanf("%s", c.phoneno);
    fseek(fp1, 0, SEEK_END); // Move to end of file before adding new record
    fwrite(&c, sizeof(c), 1, fp1);
    printf("\nRecord added successfully\n");
}

void deleterecord(FILE *fp1, char *fn) {
    FILE *temp = fopen("temp.bin", "wb");
    if (temp == NULL) {
        printf("Cannot open temporary file\n");
        return;
    }

    struct cust c;
    char name[20];
    int found = 0;
    printf("\nEnter customer name to delete: ");
    scanf("%s", name);
    rewind(fp1);

    while (fread(&c, sizeof(c), 1, fp1)) {
        if (strcmp(c.name, name) != 0) {
            fwrite(&c, sizeof(c), 1, temp);
        } else {
            found = 1;
        }
    }

    fclose(fp1);
    fclose(temp);
    remove(fn);
    rename("temp.bin", fn);
    fp1 = fopen(fn, "rb+");
    if (fp1 == NULL) {
        printf("Error reopening the file\n");
        return;
    }

    if (found) {
        printf("Record deleted successfully.\n");
    } else {
        printf("Record not found.\n");
    }
}

void listcustomer(FILE *fp1) {
    struct cust c;
    rewind(fp1); // Reset file pointer to the start
    printf("Customer List:\n");
    while (fread(&c, sizeof(c), 1, fp1)) {
        printf("Name: %s, Phone: %s\n", c.name, c.phoneno);
    }
}

int main() {
    FILE *fp1;
    char fn[] = "cust.bin";
    fp1 = fopen(fn, "rb+");
    if (fp1 == NULL) {
        fp1 = fopen(fn, "wb+");
        if (fp1 == NULL) {
            printf("Cannot open file\n");
            exit(1);
        }
    }

    int choice;
    do {
        printf("\nMenu:\n");
        printf("1. Determine the telephone number of a specific customer\n");
        printf("2. Determine the customer whose telephone number is specified\n");
        printf("3. Add a new record\n");
        printf("4. Delete a record\n");
        printf("5. List all customers\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                searchcustomer(fp1);
                break;
            case 2:
                searchtelephone(fp1);
                break;
            case 3:
                addrecord(fp1);
                break;
            case 4:
                deleterecord(fp1, fn);
                break;
            case 5:
                listcustomer(fp1);
```

```
                break;
            case 6:
                break;
            default:
                printf("Invalid choice! Please enter a number between 1 and 6.\n");
        }
    } while (choice != 6);

    fclose(fp1);
    return 0;
}
```

```
O/P: Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
Enter your choice: 5
Customer List:
Name: b, Phone: 23
Name: c, Phone: 45
Name: d, Phone: 33

Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
```

```
Enter your choice: 1
Enter customer name: d
Customer telephone is 33

Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
Enter your choice: 2
Enter phone number: 33
Customer Name is d

Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
Enter your choice: 3

Enter customer name: e

Enter customer phone number: 56

Record added successfully

Menu:
```

```
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
Enter your choice: 4

Enter customer name to delete: d
Record deleted successfully.

Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
Enter your choice: 5
Customer List:
Name: b, Phone: 23
Name: c, Phone: 45
Name: e, Phone: 56

Menu:
1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. List all customers
6. Exit
```

Name:Ashnit Kaur Bagga

ROLL NO:02

Course:PGDCSA:

SUBJECT:FUNDAMENTAL OF PROGRAMMING

ASSIGNMENT 3:

25. 10.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 features

a. Add a new record.

b. Delete a record.

c. Modify an existing record.

d. Retrieve and display an entire record for a given
ID/Name.

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

f. End of the computation/Exit.

I/P:

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

struct Employee {
   int id;
   char name[50];
   char address[100];
   char phone[15];
};

// Function prototypes
void addRecord(FILE *fp);
void deleteRecord(FILE *fp);
void modifyRecord(FILE *fp);
void retrieveRecord(FILE *fp);
void listRecords(FILE *fp);

int main() {
    FILE *fp;
    int choice;
    char filename[] = "employee.dat";

    fp = fopen(filename, "rb+");
    if (fp == NULL) {
        fp = fopen(filename, "wb+");
        if (fp == NULL) {
            printf("Cannot open file.\n");
            exit(1);
        }
    }

    do {
        printf("\nMenu:\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 an entire record for a given ID/Name\n");
        printf("5. Generate a complete list of all employee names, addresses, and telephone
numbers\n");
        printf("6. End of computation/Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
              addRecord(fp);
              break;
            case 2:
              deleteRecord(fp);
              break;
            case 3:
              modifyRecord(fp);
              break;
            case 4:
              retrieveRecord(fp);
              break;
            case 5:
              listRecords(fp);
              break;
            case 6:
              fclose(fp);
              exit(0);
            default:
              printf("Invalid choice! Please enter a number between 1 and 6.\n");
        }
    } while (choice != 6);

    fclose(fp);
    return 0;
}

// Function to add a new record
void addRecord(FILE *fp) {
    struct Employee e;
    fseek(fp, 0, SEEK_END);
    printf("Enter Employee ID: ");
    scanf("%d", &e.id);
    printf("Enter Employee Name: ");
    scanf(" %[^\n]%*c", e.name);
    printf("Enter Employee Address: ");
    scanf(" %[^\n]%*c", e.address);
    printf("Enter Employee Phone: ");
    scanf(" %[^\n]%*c", e.phone);
    fwrite(&e, sizeof(struct Employee), 1, fp);
    printf("Record added successfully.\n");
}

// Function to delete a record
void deleteRecord(FILE *fp) {
    FILE *temp;
    struct Employee e;
    int id, found = 0;
    char filename[] = "employee.dat";

    temp = fopen("temp.dat", "wb");
    if (temp == NULL) {
        printf("Cannot open temporary file.\n");
        return;
    }

    printf("Enter Employee ID to delete: ");
    scanf("%d", &id);
    rewind(fp);

    while (fread(&e, sizeof(struct Employee), 1, fp)) {
        if (e.id != id) {
            fwrite(&e, sizeof(struct Employee), 1, temp);
        } else {
            found = 1;
        }
    }

    fclose(fp);
    fclose(temp);
    remove(filename);
    rename("temp.dat", filename);

    fp = fopen(filename, "rb+");
    if (found)
        printf("Record deleted successfully.\n");
    else
        printf("Record not found.\n");
}

// Function to modify an existing record
void modifyRecord(FILE *fp) {
    struct Employee e;
    int id, found = 0;
    printf("Enter Employee ID to modify: ");
    scanf("%d", &id);
    rewind(fp);

    while (fread(&e, sizeof(struct Employee), 1, fp)) {
        if (e.id == id) {
            printf("Enter new Employee Name: ");
            scanf(" %[^\n]%*c", e.name);
            printf("Enter new Employee Address: ");
            scanf(" %[^\n]%*c", e.address);
            printf("Enter new Employee Phone: ");
            scanf(" %[^\n]%*c", e.phone);
            fseek(fp, -sizeof(struct Employee), SEEK_CUR);
            fwrite(&e, sizeof(struct Employee), 1, fp);
            found = 1;
            break;
        }
    }

    if (found)
        printf("Record modified successfully.\n");
    else
        printf("Record not found.\n");
}

// Function to retrieve and display an entire record for a given ID/Name
void retrieveRecord(FILE *fp) {
    struct Employee e;
    int id, choice, found = 0;
    char name[50];

    printf("Retrieve by:\n1. ID\n2. Name\nEnter your choice: ");
    scanf("%d", &choice);
    rewind(fp);

    if (choice == 1) {
        printf("Enter Employee ID: ");
        scanf("%d", &id);

        while (fread(&e, sizeof(struct Employee), 1, fp)) {
            if (e.id == id) {
                printf("\nEmployee ID: %d\n", e.id);
                printf("Employee Name: %s\n", e.name);
                printf("Employee Address: %s\n", e.address);
                printf("Employee Phone: %s\n", e.phone);
                found = 1;
                break;
            }
        }
    } else if (choice == 2) {
        printf("Enter Employee Name: ");
        scanf(" %[^\n]%*c", name);

        while (fread(&e, sizeof(struct Employee), 1, fp)) {
            if (strcmp(e.name, name) == 0) {
                printf("\nEmployee ID: %d\n", e.id);
                printf("Employee Name: %s\n", e.name);
                printf("Employee Address: %s\n", e.address);
                printf("Employee Phone: %s\n", e.phone);
                found = 1;
                break;
            }
        }
    } else {
        printf("Invalid choice.\n");
        return;
    }

    if (!found)
        printf("Record not found.\n");
}
```

```
// Function to generate a complete list of all employee names, addresses, and telephone numbers
void listRecords(FILE *fp) {
    struct Employee e;
    rewind(fp);

    printf("\nComplete list of employees:\n");
    while (fread(&e, sizeof(struct Employee), 1, fp)) {
        printf("\nEmployee ID: %d\n", e.id);
        printf("Employee Name: %s\n", e.name);
        printf("Employee Address: %s\n", e.address);
        printf("Employee Phone: %s\n", e.phone);
    }
}
```

```
O/P: Menu:
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display an entire record for a given ID/Name
5. Generate a complete list of all employee names, addresses, and telephone numbers
6. End of computation/Exit
Enter your choice: 5

Complete list of employees:


Employee ID: 3
Employee Name: d
Employee Address: ert
Employee Phone: 456


Employee ID: 1
Employee Name: dfg
Employee Address: rty
Employee Phone: 34

Employee ID: 2
Employee Name: RIYAA
Employee Address: Surat
Employee Phone: 789

Employee ID: 5
Employee Name: N
Employee Address: AHMD
Employee Phone: 89

Employee ID: 1
Employee Name: a
Employee Address: ahmd
Employee Phone: 45

Menu:
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display an entire record for a given ID/Name
5. Generate a complete list of all employee names, addresses, and telephone numbers
6. End of computation/Exit
Enter your choice: 1
Enter Employee ID: 6
Enter Employee Name: er
Enter Employee Address: fgh
Enter Employee Phone: 67
```

```
Record added successfully.

Menu:
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display an entire record for a given ID/Name
5. Generate a complete list of all employee names, addresses, and telephone numbers
6. End of computation/Exit
Enter your choice: 3
Enter Employee ID to modify: 6
Enter new Employee Name: y
Enter new Employee Address: tyu
Enter new Employee Phone: 78
Record modified successfully.

Menu:
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display an entire record for a given ID/Name
5. Generate a complete list of all employee names, addresses, and telephone numbers
6. End of computation/Exit
Enter your choice: 5

Complete list of employees:

Employee ID: 3
Employee Name: d
Employee Address: ert
Employee Phone: 456
Employee ID: 1
Employee Name: dfg
Employee Address: rty
Employee Phone: 34

Employee ID: 2
Employee Name: RIYAA
Employee Address: Surat
Employee Phone: 789

Employee ID: 5
Employee Name: N
Employee Address: AHMD
Employee Phone: 89

Employee ID: 1
Employee Name: a
Employee Address: ahmd
Employee Phone: 45

Employee ID: 6
Employee Name: y
Employee Address: tyu
Employee Phone: 78

Menu:
1. Add a new record
2. Delete a record
3. Modify an existing record
4. Retrieve and display an entire record for a given ID/Name
```

5. Generate a complete list of all employee names, addresses, and telephone numbers

6. End of computation/Exit

Enter your choice: