Exercise 1

A1 : Simple C Programs using I/O statements and expressions

1) **Add, subtract, multiply and divide two integers by getting inputs from the user.**

Code:

```
#include<stdio.h>

void  main()
{
    int a,b,sum,dif,mul,q;
    printf("enter two numbers");
    scanf("%d%d",&a,&b);
    sum=a+b;
    dif=a-b;
    mul=a*b;
    q=a/b;
    printf("sum of two numbers = %d\n",sum);
    printf("difference of two numbers = %d\n",dif);
    printf("product of two numbers = %d\n",mul);
    printf("quotient of two numbers = %d",q);

}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc add.c -o add

PS D:\01-College\c programs\ex1> ./add

enter two numbers2 3

sum of two numbers = 5

difference of two numbers = -1

product of two numbers = 6

quotient of two numbers = 0
```

2) Swap the values of two variables using a temporary variable.

Code:

```
#include<stdio.h>
void main()
{
    int a,b,temp;
    printf("enter two numbers for swapping");
    scanf("%d%d",&a,&b);
    temp=a;
    a=b;
    b=temp;
    printf("numbers after swapping are: %d,%d",a,b);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc swap.c -o swap

PS D:\01-College\c programs\ex1> ./swap

enter two numbers for swapping2 4

numbers after swapping are: 4,2
```

3) Print the last digit of an integer.

Code:

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

```
void main()
{
    int a;
    printf("enter number ");
    scanf("%d",&a);
    printf("the last digit of the number is %d",a%10);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc lastdig.c -o lastdig
PS D:\01-College\c programs\ex1> ./lastdig
enter number 5678
the last digit of the number is 8
```

4) Get the marks for five subjects and compute the total and average.

Code:

```
#include<stdio.h>
void main()
{
    int m1,m2,m3,m4,m5,tot=0;
    float avg=0.0;
    printf("enter 5 subject marks ");
    scanf("%d %d %d %d %d",&m1,&m2,&m3,&m4,&m5);
    tot=(m1+m2+m3+m4+m5);
    avg=tot/5;
    printf("total and average are %d and %f",tot,avg);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc totavg.c -o totavg

PS D:\01-College\c programs\ex1> ./totavg

enter 5 subject marks 89 99 97 68 89

total and average are 442 and 88.000000
```

5) Find the area of rectangle, triangle and circle by getting inputs from the user.

5) **Code:**

```
#include<stdio.h>
void main()
{
    int op;
    printf("1.rectangle\n2.triangle\n3.circle\n");
    printf("enter option\n");
    scanf("%d",&op);
    if(op==1)
    {
        float l,b;
        printf("enter l and b for rectangle\n");
        scanf("%f%f",&l,&b);
        printf("area of rectangle=%f",l*b);
    }
    else if(op==2)
    {
        float base,h;
        printf("enter b and h for triangle\n");
        scanf("%f%f",&base,&h);
        printf("area of triangle=%f",0.5*base*h);
    }
    else if(op==3)
    {
        float r;
        printf("enter radius of circle\n");
        scanf("%f",&r);
        printf("area of circle=%f",3.14*r*r);
    }
    else
    printf("invalid input");
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc area.c -o area

PS D:\01-College\c programs\ex1> ./area

1.rectangle

2.triangle

3.circle

enter option

1

enter l and b for rectangle

2 4

area of rectangle=8.000000
```

```
PS D:\01-College\c programs\ex1> gcc area.c -o area

PS D:\01-College\c programs\ex1> ./area

1.rectangle

2.triangle

3.circle

enter option

2

enter b and h for triangle

3 5

area of triangle=7.500000
```

```
PS D:\01-College\c programs\ex1> gcc area.c -o area

PS D:\01-College\c programs\ex1> ./area

1.rectangle

2.triangle

3.circle

enter option

3

enter radius of circle

5,4

area of circle=78.500000
```

6) Get values for *a, b, c and d* from the user and evaluate the expression *a*b+c^d.*

Code:

```c
#include<stdio.h>
#include<math.h>
void main()
{
    float a,b,c,d,e;
    printf("enter 4 numbers");
    scanf("%f %f %f %f",&a,&b,&c,&d);
    e=a*b+pow(c,d);
    printf("a*b+c^d=%f",e);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc pow.c -o pow

PS D:\01-College\c programs\ex1> ./pow

enter 4 numbers2.0 3.4 5.2 6.1

a*b+c^d=23320.984375
```

7) Calculate Simple Interest.

7) Code:

```c
#include<stdio.h>
void main()
{
    int p,t;
    float r,s;
    printf("enter principle amount,rate and time");
    scanf("%d %f %d",&p,&r,&t);
    s=p*r*t/100.0;
    printf("simple interest=%f",s);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc si.c -o si

PS D:\01-College\c programs\ex1> ./si
```

```
enter principle amount,rate and time20000 7.5 2

simple interest=3000.000000
```

8) Find the net salary of an employee by getting the basic pay (BP) as input.
Compute the pay
based upon the following formulae:
DA = 88% of BP
HRA = 8% of BP
CCA = Rs. 1000
Insurance = Rs. 2000
PF = 10% of BP
Gross Pay = BP + DA + HRA + CCA
Deductions = Insurance + PF
Net Pay = Gross Pay – Deduction

8) **Code:**

```
#include<stdio.h>
void main()
{
    int bp,cca=1000,in=2000;
    float da,hra,pf,gp,ded,np;
    printf("enter the basic pay of employee");
    scanf("%d",&bp);
    da=0.88*bp;
    hra=0.08*bp;
    pf=0.1*bp;
    gp=bp+da+hra+cca;
    ded=in+pf;
    np=gp-ded;
    printf("net pay of employee = %f",np);
}
```

Output:

```
PS D:\01-College\c programs\ex1> gcc bp.c -o bp

PS D:\01-College\c programs\ex1> ./bp


enter the basic pay of employee20000

net pay of employee = 36200.000000
```

Exercise 2

A2 : C Programs using I/O statements, conditional and looping constructs

1) Check whether the given integer is odd or even

Code:

```
#include<stdio.h>
void main()
{
    int a;
    printf("enter number");
    scanf("%d",&a);
    if(a%2==0)
    printf("even");
    else
    printf("odd");
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc oddeven.c -o oddeven

PS D:\01-College\c programs\ex2> ./oddeven

enter number49

odd

enter number34
```

```
even
```

2) Modify (1) to set a flag to 1 if number is odd; 0 if even and print the value of flag. (Use *conditional operator*)

2)

Code:

```
#include<stdio.h>
void main()
{
    int a,flag;
    printf("enter number\n");
    scanf("%d",&a);
    if(a%2==0)
    {
        flag=0;
    }
    else
    {
        flag=1;
    }
    printf("flag is %d",flag);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc oddevenmod.c -o oddevenmod

PS D:\01-College\c programs\ex2> ./eddevenmod

enter number34

flag is 0
```

```
enter number79

flag is 1
```

3) A company decides to give bonus to all its employees on Diwali. A 5% bonus on salary is given to the male workers and 10% bonus on salary to the female workers. If the salary of the employee is less than Rs. 10000/- then the employee gets an extra 2% bonus on salary. Calculate the bonus that the employee will get and also display the total salary.

3) **Code:**

```c
#include<stdio.h>
void main()
{
    int bsal;
    char gender;
    float nsal,bonus;
    printf("enter employee gender m for male and f for female");
    scanf("%s",&gender);
    printf("enter basic salary of employee");
    scanf("%d",&bsal);
    if(gender=='m')
    {
     if(bsal<10000)
     bonus=0.07*bsal;
     else
     bonus=0.05*bsal;
    }
    else
    {
     if(bsal<10000)
     bonus=0.12*bsal;
     else
     bonus=0.10*bsal;
    }
    printf("bonus = %f",bonus);
    nsal=bsal+bonus;
    printf("total salary = %f",nsal);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc bonus.c -o bonus

PS D:\01-College\c programs\ex2> ./bonus

enter employee gender m for male and f for femalef

enter basic salary of employee12000

bonus = 1200.000000

total salary = 13200.000000
```

```
PS D:\01-College\c programs\ex2> gcc bonus.c -o bonus

PS D:\01-College\c programs\ex2> ./bonus

enter employee gender m for male and f for femalef

enter basic salary of employee1200

bonus = 144.000000

total salary = 1344.000000
```

```
PS D:\01-College\c programs\ex2> gcc bonus.c -o bonus

PS D:\01-College\c programs\ex2> ./bonus

enter employee gender m for male and f for femalem

enter basic salary of employee1200

bonus = 84.000000

total salary = 1284.000000
```

```
PS D:\01-College\c programs\ex2> gcc bonus.c -o bonus
PS D:\01-College\c programs\ex2> ./bonus

enter employee gender m for male and f for femalem

enter basic salary of employee12000

bonus = 600.000000

total salary = 12600.000000
```

4) Let the user enter a whole number *N* between 3 and 10 and print an egg timer of size *N*.
Validate *N* to be non-zero positive number.

```
Example
Enter a number : 4
*******
*****
***
*
***
*****
*******
```

4) **Code:**

```
#include<stdio.h>
void main()
    {
    int n,i;
    printf("Enter a number between 3 and 10\n");
    scanf("%d",&n);

    for(int j=n;j>0;j--)
        {
        for(int k=0;k<n-i;k++)
            printf(" ");

        for(int i=2*j-1;i>0;i--)
            printf("*");
        printf("\n");
        }
    for(int j=2;j<=n;j++)
        {
        for(int k=n-i;k>0;k--)
            printf(" ");

        for(int i=1;i<=2*j-1;i++)
            printf("*");
        printf("\n");
        }
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc eggtimer.c -o eggtimer

PS D:\01-College\c programs\ex2> ./eggtimer

enter a number between 3 and 104

   *******

     *****

      ***

        *

      ***

    *****

  *******
```

5) Compute the sum of *N* integers. (Use *do-while*) (Version 1)
a. Get input for *N*, multiple times until -999 is given (Version 2)
b. Get input for *N*, multiple times until ‘Y’ is given (Version 3)
c. Validate *N* to be a positive number less than 100. Print error message for invalid input and exit. (Use *break*) (Version 4)
d. If input is invalid, print message and ask for another input. (Version 5)

5a) **Code:**

```
#include<stdio.h>
void main()
{
int n,sum;
sum=0;
do
{
printf("Enter number:");
scanf(" %d",&n);
if(n!=-999)
sum+=n;
}while(n!=-999);
printf("SUM: %d",sum);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc 5a.c -o 5a

PS D:\01-College\c programs\ex2> ./5a

Enter number:234

Enter number:456

Enter number:789

Enter number:-
999

SUM: 1479
```

5b)Code:

```c
#include<stdio.h>
void main()
{
int sum,x;
char ch;
x=0;
sum=0;
ch='X';
do
{
printf("Enter Y to terminate, anything else to continue:");
scanf(" %c",&ch);
if(ch!='Y')
{
printf("Enter a number");
scanf("%d",&x);
sum=sum+x;
}

}while(ch!='Y');
printf("SUM: %d",sum);
}
```

```
PS D:\01-College\c programs\ex2> gcc 5b.c -o 5b

PS D:\01-College\c programs\ex2> ./5b

Enter Y to terminate, anything else to continue:t
Enter a number345

Enter Y to terminate, anything else to continue:r

Enter a number346

Enter Y to terminate, anything else to continue:e

Enter a number658

Enter Y to terminate, anything else to continue:o

Enter a number67

Enter Y to terminate, anything else to continue:y

Enter a number67

Enter Y to terminate, anything else to continue:Y

SUM: 1483
```

5c) **Code:**

```
#include<stdio.h>
void main()
{
int n,sum;
sum=0;
do
{
printf("Enter number:");
scanf(" %d",&n);
if(n<=100)
sum+=n;
else
printf("error in input\n");
}while(n<=100);
printf("SUM: %d",sum);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc 5c.c -o 5c

PS D:\01-College\c programs\ex2> ./5c

Enter number:23
Enter number:34

Enter number:67

Enter number:89

Enter number:21

Enter number:56

Enter number:109

error in input

SUM: 290
```

5d) **Code:**

```c
#include<stdio.h>
void main()
{
int n,sum;
sum=0;
do
{
printf("Enter number:");
scanf(" %d",&n);
if(n<=100)
sum+=n;
else
{
printf("error in input\nenter another number");
scanf("%d",&n);
sum+=n;
break;
}
}while(n<=100);
printf("SUM: %d",sum);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc 5d.c -o 5d

PS D:\01-College\c programs\ex2> ./5d
```

```
Enter number:34

Enter number:45

Enter number:56

Enter number:89

Enter number:108

error in input

enter another number56

SUM: 280
```

6) Design a calculator to perform the operations namely addition, subtraction, multiplication, division and square of a number. (Note: Provide operation options for the user to choose, after getting two numbers of type *float*. Let the calculator performs its operations till the user wishes.) (Use *case*)

6) **Code:**

```
#include <stdio.h>

void main()
{
        char yes;
        float a, b, c;
    int choice;

        yes = 'y';
        while(yes == 'y' || yes == 'Y')
        {
        printf("Enter first integer: ");
        scanf("%f", &a);

        printf("Enter second integer: ");
        scanf("%f", &b);
    printf("\n1.Add, 2.Subtract,3.Multiply,4.divide");
        scanf("%d", &choice);

        printf("\n");
        switch(choice)
        {
                case(1):
                    c = a + b;
                    printf("%f + %f = %f\n", a, b, c);
                    break;
                case(2):
                    c = a - b;
                    printf("%f - %f = %f\n", a, b, c);
                    break;
                case(3):
                    c = a * b;
                    printf("%f * %f = %f\n", a, b, c);
                    break;
                case(4):
                    c = a / (float)b;
                    printf("%f / %f = %f\n", a, b, c);
                    break;
                default:
                    printf("Invalid,try again.\n");
        }

        printf("\nAgain (Y/N): ");
        scanf(" %c", &yes);
    }
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc calc.c -o calc

PS D:\01-College\c programs\ex2> ./calc

Enter first integer: 23

Enter second integer: 45

1.Add, 2.Subtract,3.Multiply,4.divide1

23.000000 + 45.000000 = 68.000000

Again (Y/N): y

Enter first integer: 23
Enter second integer: 67

1.Add, 2.Subtract,3.Multiply,4.divide2

23.000000 - 67.000000 = -44.000000

Again (Y/N): YY

Enter first integer:23

 Enter second integer: 67

1.Add, 2.Subtract,3.Multiply,4.divide

23.000000 - 67.000000 = -44.000000

Again (Y/N): Enter first integer: 25

Enter second integer: 54

1.Add, 2.Subtract,3.Multiply,4.divide3

25.000000 * 54.000000 = 1350.000000

Again (Y/N): y

Enter first integer: 45

Enter second integer: 98

1.Add, 2.Subtract,3.Multiply,4.divide4

45.000000 / 98.000000 = 0.459184

Again (Y/N): n
```

7) Check if a number has three consecutive 5s. If yes, print YES, else print NO.
Example
Number: 1353554
Result: NO
Number: 345559
Result: YES

7) **Code:**

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

```
{
int no,x;
int n1,n2,n3;
printf("ENTER NUMBER:");
scanf("%d",&no);
n1=no%10;
n2=(no%100-n1)/10;
n3=(no%1000-n2)/100;
x=0;
while(n3!=0)
{
if(n1==5 && n2==5 && n3==5)
{
printf("YES");
x=1;
break;
}
else
{
no=no/10;
n1=no%10;
n2=(no%100-n1)/10;
n3=(no%1000-n2)/100;
}
}
if(x==0)
printf("NO");
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc cons.c -o cons

PS D:\01-College\c programs\ex2> ./cons

ENTER NUMBER:45556

YES


PS D:\01-College\c programs\ex2> gcc cons.c -o cons

PS D:\01-College\c programs\ex2> ./cons

ENTER NUMBER:45367

NO
```

8) Calculate the parking charges of a vehicle. Enter the type of vehicle (as a character for eg. B for Bus). Read the hours and minutes when the vehicle enters the parking lot and also when it leaves. Calculate the difference between the two timings to calculate the number of hours and minutes for which the vehicle was parked. Finally calculate the parking charges based on the following rules and display the result.

Rate till 3 hours Rate after 3 hours
· Truck/Bus Rs. 20/hour Rs. 30/hour
· Car Rs. 10/hour Rs. 20/hour
· Scooter/Cycle/Motor Cycle Rs. 5/hour Rs. 10/hour

8) **Code:**

```
#include<stdio.h>
void main()
{
int charge;
int hrs,mins,hrs1,mins1,hrsfin,minsfin,totminsin,totminsout,totmins;
char vehi;
printf("Enter the Vehicle type: B for bus/truck,C for Car,S for scooter/bike"
);
scanf("%c",&vehi);
printf("Enter in time hrs and mins");
scanf("%d%d",&hrs,&mins);
printf("Enter out time hrs and mins");
scanf("%d%d",&hrs1,&mins1);
totminsin=hrs*60+mins;
totminsout=hrs1*60+mins1;
totmins=totminsout-totminsin;
hrsfin=totmins/60;
minsfin=totmins%60;
if(hrsfin<=3)
{
if(vehi=='B')
charge=20*hrsfin;
else if(vehi=='C')
charge=10*hrsfin;
else if(vehi=='S')
charge=5*hrsfin;
}
else
{
if(vehi=='B')
charge=20*3+(hrsfin-3)*30;
else if(vehi=='C')
charge=10*3+(hrsfin-3)*20;
else if(vehi=='S')
charge=5*3+(hrsfin-3)*10;
}
printf("CHARGE:%d",charge);
}
```

Output:

```
PS D:\01-College\c programs\ex2> gcc parking.c -o parking

PS D:\01-College\c programs\ex2> ./parking

Enter the Vehicle type: B for bus/truck,C for Car,S for scooter/bikeC

Enter in time hrs and mins2 30

Enter out time hrs and mins4 40

CHARGE:20
```

```
PS D:\01-College\c programs\ex2> gcc parking.c -o parking

PS D:\01-College\c programs\ex2> ./parking

Enter the Vehicle type: B for bus/truck,C for Car,S for scooter/bikeB

Enter in time hrs and mins3 00

Enter out time hrs and mins6 20

CHARGE:60
```

```
PS D:\01-College\c programs\ex2> gcc parking.c -o parking
PS D:\01-College\c programs\ex2> ./parking

Enter the Vehicle type: B for bus/truck,C for Car,S for scooter/bikeS

Enter in time hrs and mins2 30

Enter out time hrs and mins9 30

CHARGE:55
```

Exercise 3

A3 : User-defined Functions

1. Define a function *CheckOddEven(num)* that checks if the num is odd or even; sets a flag accordingly and return it. Use this function to find the sum of even and odd numbers in a given input of *N* numbers.

1)Code:

```
#include<stdio.h>

int checkoddeven(int n)
{
  int flag;
  if(n%2==0)
    {
      flag=1;
     }
   else
    flag=0;
   if(flag==1)
   {
    printf("\n the number is even");
    }
   else
    {

    printf("\n the number is odd");
   }
 return(flag);
}

void  main()
{
  int x,sumo,sume,y;char c;
  sumo=0;
  sume=0;
  do
 {
   printf("\n enter a number");
  scanf("%d",&x);
  y=checkoddeven(x);
  if(y==1)
  sume=sume+x;
  else
  sumo=sumo+x;
  printf("\n do you want to continue??");
  scanf(" %c",&c);
 }while(c=='y');
 printf("\n even sum is %d",sume);
 printf("\n odd sum is %d\n",sumo);
}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc oddeven.c -o oddeven

PS D:\01-College\c programs\ex3> ./oddeven

 enter a number34

 the number is even

 do you want to continue??y

 enter a number23

 the number is odd

 do you want to continue??y

 enter a number67

 the number is odd

 do you want to continue??y

 enter a number45

 the number is odd

 do you want to continue??y

 enter a number21

 the number is odd

 do you want to continue??n

 even sum is 34

 odd sum is 156
```

2. Write a C function *ReverseNum(num)* that takes integer *num* and reverses its digits.
Let num be passed by reference.
Input:
453275
Output:
572354

2) **Code:**

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

```
int reversenum(int *n)
{   int newn,d;
    newn=0;
    while(*n>0)
    {   d=*n%10;
        *n=*n/10;
        newn=(newn*10)+d;
    }
    return newn;
}
void main()
{   int num,rev;
    printf("Enter number ");
    scanf("%d",&num);
    rev=reversenum(&num);
    printf("\nReverse of the number is %d\n",rev);

}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc revnum.c -o revnum

PS D:\01-College\c programs\ex3> ./revnum

Enter number 2345

Reverse of the number is 5432
```

**3. Write a function *power(X,N)* that will allow a floating-point number to be raised to an
integer power and return the result. Y = X
N .
In other words, evaluate the formula where y and x are floating-point variables and n
is an integer variable. Write a C program that will read in numerical values for x and n,
evaluate the formula using *power(X,N)* and then display the calculated result.**

3) **Code:**

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

```
float power(float x,int n)
{   float y=1;
    for(int i=1;i<=n;i++)
        y=y*x;
    return y;
}
void main()
{   int n;
    float x,y;
    printf("Enter x and n ");
    scanf("%f %d",&x,&n);
    y=power(x,n);
    printf("X^n is %f \n",y);
}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc pow.c -o pow

PS D:\01-College\c programs\ex3> ./pow

Enter x and n 23.0 4

X^n is 279841.000000
```

4. Find the product of *n* floating point numbers. The numbers should be read from the keyboard. No looping construct should be used. (Hint: Use recursion)

4) **Code:**

```
#include<stdio.h>

float prod(int n)
{   if(n==0)
        return 1;
    else
    {   float num;
        printf("Enter number ");
        scanf("%f",&num);
        return num*prod(n-1);
    }
}
void main()
{
```

```
int n;
printf("Enter n value ");
scanf("%d",&n);
float p=prod(n);
printf("Product of numbers is %f",p);

}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc prdt.c -o prdt

PS D:\01-College\c programs\ex3> ./prdt

Enter n value 6

Enter number 23

Enter number 45

Enter number 34.0

Enter number 21.88

Enter number 3.5

Enter number 2.0

Product of numbers is 5369994.000000
```

5. Write recursive functions for the following:
i. Reads N and prints from N to 0.
Input Output
10 9876543210
ii. Sum of digits of a given number
Input Output
34562 20

5a) **Code:**

```
#include<stdio.h>
int rev(int n)
{
 printf("%d  ",n);
  if(n==0)
    printf("\n");
  else
  rev(n-1);
}
void main()
```

```
{
 int x;
 printf("\n enter a number");
 scanf("%d",&x);
 rev(x);
}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc count.c -o count

PS D:\01-College\c programs\ex3> ./count

enter a number10

10  9  8  7  6  5  4  3  2  1  0
```

b) Code:

```
#include <stdio.h>
int sum(int n)
{   int a;
    if (n == 0)
       return 0;
    a=n%10;
    return ( a + sum(n / 10));
}
void main()
{
    int num;
    printf("\n enter a number ");
    scanf("%d",&num);
    int result = sum(num);
    printf(" sum of digits  is %d\n",result);
}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc sum.c -o sum

PS D:\01-College\c programs\ex3> ./sum

enter a number 23456

 sum of digits  is 20
```

6. Write a function in C to compute the distance between two points and use it to develop another
function that will compute the area of the triangle whose vertices are *A(x1, y1), B(x2, y2), and*

_C(x3, y3)_. Use these functions to develop a function which returns a value 1 if the point _(x, y)_ lies inside the triangle _ABC_, otherwise a value 0. Write a program in C to test the above mentioned functions.

6) Code:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
float distance(float x1 ,float y1,float x2, float y2)
{   float dist;
    float sq=pow(x2-x1,2)+pow(y2-y1,2);
    dist=pow(sq,0.5);
    return dist;
}
float ar(float s1,float s2,float s3)
{   float area,s;
    s=(s1+s2+s3)/2.0;
    area=pow(s*(s-s1)*(s-s2)*(s-s3),0.5);
    return area;
}
void check(float x1,float x2,float x3,float x4,float y1,float y2,float y3,flo
at y4)
{   float a,a1,a2,a3;
    a=ar(distance(x1,y1,x2,y2),distance(x2,y2,x3,y3),distance(x3,y3,x1,y1));
    a3=ar(distance(x1,y1,x2,y2),distance(x2,y2,x4,y4),distance(x4,y4,x1,y1));
    a2=ar(distance(x1,y1,x4,y4),distance(x4,y4,x3,y3),distance(x3,y3,x1,y1));
    a1=ar(distance(x4,y4,x2,y2),distance(x2,y2,x3,y3),distance(x3,y3,x4,y4));
    printf("%f %f %f \n",a1,a2,a3);
    if(a-(a1+a2+a3)<=0.00001 || (a1+a2+a3)-a<=0.00001)
        printf("Point lies inside triangle \n");
    else
        printf("Point lies outside triangle \n");
}
int main()
{   float x1,y1,x2,y2,x3,y3,x4,y4;
    float dist,area;
    printf("Enter point 1 ");
    scanf("%f %f",&x1,&y1);
    printf("Enter point 2 ");
    scanf("%f %f",&x2,&y2);
    dist=distance(x1,y1,x2,y2);
    printf("Distance between the points is %f \n",dist);
    printf("Enter triangle coordinates \nEnter point 1 ");
    scanf("%f %f",&x1,&y1);
    printf("Enter point 2 ");
    scanf("%f %f",&x2,&y2);
    printf("Enter point 3 ");
    scanf("%f %f",&x3,&y3);
    area=ar(distance(x1,y1,x2,y2),distance(x2,y2,x3,y3),distance(x3,y3,x1,y1)
);
    printf("Area is %f \n",area);
    printf("Enter point ");
    scanf("%f %f",&x4,&y4);
    check(x1,x2,x3,x4,y1,y2,y3,y4);
    return 1;
```

Output:

```
PS D:\01-College\c programs\ex3> gcc dist.c -o dist

PS D:\01-College\c programs\ex3> ./dist

Enter point 1 3.2 2.5

Enter point 2 3 4.  6.7 3.4

Distance between the points is 3.613862

Enter triangle coordinates

Enter point 1 2.3 1.4

Enter point 2 3.4 5.6

Enter point 3 2.7 8.4

Area is 3.009994

Enter point 2.3 4.5

1.925000 0.619999 1.705000

Point lies inside triangle
```

7. Write user-defined functions in C to convert decimal to binary and vice-versa. Test the functions by choosing the option from the menu.

8) Code:

```
#include<stdio.h>
#include<math.h>
void btd(int n)
{
int dec = 0, i = 0, rem;
while (n != 0)
{
         rem = n % 10;
         n /= 10;
        dec=dec+(rem*(pow(2,i)));
        i++;
}
printf("decimal number is : %d\n",dec);
}
void dtb(int n)
{
     int x,y=0,i=1;
     printf("the binary number is ");
     while(n>0)
    {
            x=n%2;
            y=y+(x*i);
            n=n/2;
            i=i*10;
    }
    printf("%d\n",y);
}
void main()
{
      int n,ch;
      char c;
      c='y';
     while(c=='y')
     {
           printf("MENU");
           printf("\n1)decimal to binary");
           printf("\n2)binary to decimal");
           printf("\nenter your choice");
           scanf("%d",&ch);
           switch(ch)
          {
              case 1:
              printf("enter a decimal number");
              scanf("%d",&n);
              dtb(n);
```

```
             break;
             case 2:
             printf("enter a binary number");
             scanf("%d",&n);
            btd(n);
           break;
         }
         printf("do you want to continue(y/n)? ");
         scanf(" %c",&c);
}
```

Output:

```
PS D:\01-College\c programs\ex3> gcc decbin.c -o decbin

PS D:\01-College\c programs\ex3> ./decbin

MENU

1)decimal to binary

2)binary to decimal

enter your choice1

enter a decimal number23

the binary number is 10111

do you want to continue(y/n)? y

MENU

1)decimal to binary

2)binary to decimal

enter your choice10  2

enter a binary number2 109 101100

decimal number is : 172

do you want to continue(y/n)? n
```

Exercise 4

A4 : Arrays in C

1. Write a program to read a set of integers and find the sum of positive numbers, negative numbers and zeros.

1)**Code:**

```
#include<stdio.h>
void main()
{
    int a[100],s,x,p=0,n=0,z=0;
    printf("\nEnter size of array ");
    scanf("%d",&s);
    for(int i=0;i<s;i++)
    {
        printf("\nEnter element ");
        scanf("%d",&x);
        a[i]=x;
    }
    for(int i=0;i<s;i++)
    {
        if(a[i]>0)
            p++;
        else if(a[i]<0)
            n++;
        else
            z++;
    }
    printf("\nNo. of positive integers : %d",p);
    printf("\nNo. of negative integers : %d",n);
    printf("\nNo. of zeroes : %d",z);
}
```

Output:

```
D:\01-College\c programs\arrays>gcc sposnegzero.c -o sposnegzero

D:\01-College\c programs\arrays>sposnegzero

Enter size of array 6

Enter element -9

Enter element -8

Enter element 0
Enter element 0

Enter element 8

Enter element 67

No. of positive integers : 2

No. of negative integers : 2

No. of zeroes : 2
```

2. Write user-defined functions to find the mean, mode, range, variance and the standard deviation of a set of *N* integers, and also make the elements of the array unique. Use a menu driven program to demonstrate the same. (Use functions with return type and with required arguments. No global variables to be used)

2)**Code:**

```
#include<stdio.h>
#include<math.h>
float mean(int a[],int n)
{
    float s=0.0;
    for(int i=0;i<n;i++)
    {
        s+=a[i];
    }
    s/=n;
    return s;
}
int mode(int a[],int n)
{
    int max=0,mo;
    for(int i=0;i<n;i++)
    {
        int count=0,x=a[i];
        for(int j=0;j<n;j++)
        {
            if(a[j]==x)
                count++;
        }
        if(count>max)
        {
            max=count;
            mo=x;
        }
    }
    return mo;
}
int range(int a[],int n)
{
    for(int i=1;i<n-1;i++)
        for(int j=1;j<n-1;j++)
            if(a[j]>a[j+1])
            {
                int t = a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
    return (a[n-1]-a[0]);
}
float variance(int a[],int n)
{
    float m=mean(a,n);
    float x[100],s=0.0;
    for(int i=0;i<n;i++)
    {
        x[i]=(a[i]-m)*(a[i]-m);
        s+=x[i];
    }
    s/=n;
    return s;
}

float stddev(int a[],int n)
{
    float s=pow(variance(a,n),0.5);
    return s;
}
void unique(int a[],int n)
{
    int m=n;
    for(int i=0;i<n;i++)
    {
        int x=a[i];
        for(int j=i+1;j<n;j++)
            if(a[j]==x)
            {
                a[j]=0;
                m--;
            }
    }
    int b[100],j=0;
    for(int i=0;i<n;i++)
        if(a[i]==0)
```

```
            continue;
        else
        {
            b[j]=a[i];
            j++;
        }
    for(int i=0;i<m;i++)
        printf(" %d",b[i]);

}
void main()
{
    int a[100],n,x,mo,r,c;
    float m,v,s;
    printf("\nEnter size of array ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        printf("\nEnter element ");
        scanf("%d",&x);
        a[i]=x;
    }
    printf("\nMenu : ");
    printf("\n1. Mean : ");
    printf("\n2. Mode : ");
    printf("\n3. Range : ");
    printf("\n4. Variance : ");
    printf("\n5. Standard Deviation : ");
    printf("\n6. Unique array : ");
    printf("\nEnter choice : ");
    scanf("%d",&c);
    switch(c)
    {
    case 1 :
        m=mean(a,n);
        printf("\nMean : %f",m);
        break;
    case 2 :
        mo=mode(a,n);
        printf("\nMode : %d",mo);
        break;
    case 3 :
        r=range(a,n);
        printf("\nRange : %d",r);
        break;
    case 4 :
        v=variance(a,n);
        printf("\nVariance : %f",v);
        break;
    case 5 :
        s=stddev(a,n);
        printf("\nStandard Deviation : %f",s);
        break;
    case 6 :
        unique(a,n);
        break;
    default :
        break;
    }
    printf("\n");
}
```

Output:

```
D:\01-College\c programs\arrays>gcc meanmodevarstd.c -o meanmodevarstd

D:\01-College\c programs\arrays>meanmodevarstd

Enter size of array 6

Enter element 1

Enter element 2

Enter element 3

Enter element 4

Enter element 5

Enter element 7

Menu :

1. Mean :

2. Mode :

3. Range :

4. Variance :

5. Standard Deviation :

6. Unique array :

Enter choice : 1

Mean : 3.666667
```

```
D:\01-College\c programs\arrays>meanmodevarstd

Enter size of array 6

Enter element 2

Enter element 3

Enter element 4

Enter element 4

Enter element 5

Enter element 1

Menu :

1. Mean :

2. Mode :

3. Range :

4. Variance :

5. Standard Deviation :

6. Unique array :

Enter choice : 2

Mode : 4


D:\01-College\c programs\arrays>meanmodevarstd

Enter size of array 6

Enter element 1

Enter element 2

Enter element 3

Enter element 6

Enter element 8
Enter element 5
Menu :
1. Mean :
2. Mode :
3. Range :
4. Variance :
5. Standard Deviation :
6. Unique array :
Enter choice : 3
Range : 7

D:\01-College\c programs\arrays>meanmodevarstd
Enter size of array 6
Enter element 2
Enter element 3
Enter element 5
Enter element5
Enter element 4
Enter element 7
Menu :
1. Mean :
2. Mode :
3. Range :
4. Variance :
5. Standard Deviation :
6. Unique array :
```

```
Enter choice : 4

Variance : 2.555556


D:\01-College\c programs\arrays>meanmodevarstd

Enter size of array 6

Enter element 2

Enter element 4

Enter element 6

Enter element 8

Enter element 5

Enter element 7


Menu :

1. Mean :

2. Mode :

3. Range :

4. Variance :

5. Standard Deviation :

6. Unique array :

Enter choice : 5

Standard Deviation : 1.972027


D:\01-College\c programs\arrays>meanmodevarstd

Enter size of array 6

Enter element 5

Enter element 4
Enter element 5

Enter element 3

Enter element 4

Enter element 2

Menu :

1. Mean :

2. Mode :

3. Range :

4. Variance :

5. Standard Deviation :

6. Unique array :

Enter choice : 6

 5 4 3
```

3. Write a program that accepts a set of digits (0 to 9) as input and prints a horizontal histogram representing the occurrences of each digit.

Example:

```
Enter a Number : 12
Enter 12 digits:
1,7,2,9,6,7,1,3,7,5,7,9
Histogram
0
1 * *
2 *
3 *
4
5 *
6 *
7 * * * *
8
9 * *
```

3) **Code:**

```
#include<stdio.h>
void histogram(int a[],int n)
{
    int max=a[0];
    for(int i=0;i<n;i++)
    {
        if(a[i]>max)
            max=a[i];
    }
    for(int i=0;i<=max;i++)
    {
        printf("%d",i);
        for(int j=0;j<n;j++)
            if(a[j]==i)
                printf(" *");
        printf("\n");
    }
}
void main()
{
    int a[100],n,x;
    printf("\nEnter size of array ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        printf("\nEnter element ");
        scanf("%d",&x);
        a[i]=x;
    }
    histogram(a,n);
    printf("\n");
}
```

Output:

```
D:\01-College\c programs\arrays>gcc histogram.c -o histogram

D:\01-College\c programs\arrays>histogram

Enter size of array 9

Enter element 2

Enter element 4

Enter element 6

Enter element 3

Enter element 1

Enter element 2

Enter element 3

Enter element 5

Enter element 5

0

1 *

2 * *

3 * *

4 *

5 * *

6 *
```

4. Write a program that reads and stores five quiz marks (out of 100) for each of five students.

Perform the following by user-defined functions: Compute the total score and average score for each student, and the average score, high score and low score for each quiz. Extend the program so that the deviation of each student's average from the overall class average can be

determined. Display the class average, followed by each student's individual average quiz scores, and the deviation from the class average.

4) **Code:**

```
#include<stdio.h>
void main()
{
int A[5][5];
for(int i=0;i<5;i++)
{   printf("Enter marks for student %d\n",i+1);
    for(int j=0;j<5;j++)
    {   printf("Quiz %d\t",j+1);
        scanf("%d",&A[i][j]);
    }
}
int T[5]={0,0,0,0,0};
float a[5];
for(int i=0;i<5;i++)
{   for(int j=0;j<5;j++)
        T[i]+=A[i][j];
    a[i]=T[i]/5;
}
float avg[5]={0,0,0,0,0};
int h[5],l[5];
for(int j=0;j<5;j++)
{   h[j]=A[0][j];
    l[j]=A[0][j];
    for(int i=0;i<5;i++)
    {   avg[j]+=A[i][j];
        if(h[j]<A[i][j])
            h[j]=A[i][j];
        if(l[j]>A[i][j])
            l[j]=A[i][j];
    }
    avg[j]/=5;
}
float CAV;
for(int i=0;i<5;i++)
    CAV+=avg[i];
CAV/=5;
float dev[5];
for(int i=0;i<5;i++)
    dev[i]=a[i]-CAV;
printf("CLASS AVERAGE IS %f\n",CAV);
for(int i=0;i<5;i++)
    printf("Student %d Average in all Quizzes - %f\n",i+1,a[i]);
for(int i=0;i<5;i++)
```

```
    printf("Student %d Deviation from Class average - %f\n",i+1,dev[i]);
}
```

Output:

```
D:\01-College\c programs\arrays>gcc quiz.c -o quiz

D:\01-College\c programs\arrays>quiz

Enter marks for student 1

Quiz 1  90

Quiz 2  89

Quiz 3  99

Quiz 4  79

Quiz 5  70

Enter marks for student 2

Quiz 1  99

Quiz 2  98

Quiz 3  97

Quiz 4  67

Quiz 5  88

Enter marks for student 3

Quiz 1  90

Quiz 2  90

Quiz 3  98

Quiz 4  78

Quiz 5  89

Enter marks for student 4
```

```
Quiz 1  67

Quiz 2  89

Quiz 3  90

Quiz 4  89

Quiz 5  78

Enter marks for student 5

Quiz 1  67

Quiz 2  90

Quiz 3  99

Quiz 4  86

Quiz 5  84

CLASS AVERAGE IS 86.400002

Student 1 Average in all Quizzes - 85.000000

Student 2 Average in all Quizzes - 89.000000

Student 3 Average in all Quizzes - 89.000000

Student 4 Average in all Quizzes - 82.000000

Student 5 Average in all Quizzes - 85.000000

Student 1 Deviation from Class average - -1.400002

Student 2 Deviation from Class average - 2.599998

Student 3 Deviation from Class average - 2.599998

Student 4 Deviation from Class average - -4.400002

Student 5 Deviation from Class average - -1.400002
```

5. Implement the children's hand game, Rock-paper-scissors. Rock Paper Scissors is a two player game. Each player chooses one of rock, paper or scissors, without knowing the other player's choice. The winner is decided by a set of rules:

· Rock's strength is doubled (temporarily) when fighting scissors, but halved

(temporarily) when fighting paper.

· In the same way, paper has the advantage against rock, and scissors against paper

If both players choose the same thing, there is no winner for that round. For this task, the computer will be one of the players. Let the computer choose randomly. Make 10 rounds of choice, display the score and winner.

5) **Code:**

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int p,c,s1=0,s2=0,n=1;
    printf("\n1. Rock ");
    printf("\n2. Paper ");
    printf("\n3. Scissor ");
    while(n<=10)
    {
        printf("\nEnter player's choice ");
        scanf("%d",&p);
        c=rand()%3;
        switch(p)
        {
        case 1 :
            if(c==1)
                n++;
            else if(c==2)
            {
                s2++;
                n++;
            }
            else
            {
                s1++;
                n++;
            }
            break;
        case 2 :
            if(c==2)
                n++;
            else if(c==3)
            {
                s2++;
                n++;
            }
            else
            {
                s1++;
                n++;
            }
            break;
        case 3 :
            if(c==3)
                n++;
            else if(c==1)
            {
                s2++;
                n++;
            }
            else
            {
                s1++;
                n++;
            }
            break;
        default :
            break;
        }
    }
    if(s1>s2)
        printf("\nPlayer wins !!! ");
    else
        printf("\nComputer wins !!! ");
    printf("\n");
}
```

Output:

```
D:\01-College\c programs\arrays>gcc rockpaper.c -o rockpaper

D:\01-College\c programs\arrays>rockpaper

1. Rock
2. Paper
3. Scissor
Enter player's choice 1

Enter player's choice 2

Enter player's choice 3

Enter player's choice 2

Enter player's choice 1
Enter player's choice 3

Enter player's choice 2

Enter player's choice 1

Enter player's choice 3

Enter player's choice 2

Computer wins !!!
```

Exercise 5- Strings

1) Implement the following as user-defined functions.

a. strcat(str1, str2)   Appends str2 to str1

Code:

```
#include<stdio.h>
char strcon(char str1[],char str2[])
{
    int i,j,l=0,m=0;
    for(i=0;str1[i]!='\0';i++)
    {
        l++;
    }
    for(j=0;str1[j]!='\0';j++)
    {
        m++;
    }
    for(i=0;i<m;i++)
    {
        str1[l+i]=str2[i];
    }
    str1[l+m]='\0';
}
void main()
{
    char str1[100],str2[100];
    printf("enter the strings\n");
    printf("1. ");
    gets(str1);
    printf("2. ");
    gets(str2);
    strcon(str1,str2);
    printf("The concatenated string is %s",str1);
}
```

Output:

```
D:\01-College\c programs\strings>gcc strcat.c -o strcat
D:\01-College\c programs\strings>strcat
enter the strings
1. hello
2. world
The concatenated string is helloworld
```

b.strncpy(dest, src, n) Copies up to n characters from src to dest string

<u>Code:</u>

```
#include<stdio.h>
void strncpy(char dest[],char src[],int n)
{
    for(int i=0;i<n;i++)
    {
        dest[i]=src[i];
    }
    dest[n]='\0';
}
void main()
{
    char dest[100],src[100];
    int n;
    printf("enter the string to be copied: ");
    gets(src);
    printf("enter the number of characters to be copied: ");
    scanf("%d",&n);
    strncpy(dest,src,n);
    printf("The copied string is %s",dest);
}
```

<u>Output:</u>

```
D:\01-College\c programs\strings>gcc strncpy.c -o strncpy
D:\01-College\c programs\strings>strncpy
enter the string to be copied: helloworld
enter the number of characters to be copied: 5
The copied string is hello
```

c. strchr(str1, ch)   Scans the string str1 for the first occurrence of the character ch and returns the position

Code:

```
#include<stdio.h>
int strchr(char str1[],char ch)
{
    int pos;
    for(int i=0;str1[i]!=0;i++)
    {
        if(str1[i]==ch)
        {
            pos=i+1;
            break;
        }
    }
    return pos;
}
void main()
{
    int pos;
    char str1[100],ch;
    printf("enter the string: ");
    gets(str1);
    printf("enter the character: ");
    ch=getchar();
    pos=strchr(str1,ch);
    printf("The position of the first occurence of the character %c in %s is
%d",ch,str1,pos);
}
```

Output:

```
D:\01-College\c programs\strings>gcc strchr.c -o strchr
D:\01-College\c programs\strings>strchr
enter the string: helloworld
enter the character: l
The position of the first occurence of the character l in helloworld is 3
```

d.strset(str1, ch)   Sets all characters in the string str1 to the character ch

Code:

```
#include<stdio.h>
void strset(char str1[],char ch)
{
    for(int i=0;str1[i]!='\0';i++)
    {
        str1[i]=ch;
    }
}
void main()
{
    char str1[100],ch;
    printf("Enter the string: ");
    gets(str1);
    printf("Enter character to replace all the characters in the string: ");
    ch=getchar();
    strset(str1,ch);
    printf("The substituted string is %s",str1);
}
```

Output:

```
D:\01-College\c programs\strings>gcc strset.c -o strset
D:\01-College\c programs\strings>strset
Enter the string: helloworld
Enter character to replace all the characters in the string: h
The substituted string is hhhhhhhhhh
```

e. strcmpi(str1, str2)   Compares str1 and str2 ignoring the case sensitivity and return

<u>Code:</u>

```
#include<stdio.h>
#include<ctype.h>
int strcmpi(char str1[],char str2[])
{
    int i,j;
    for(i=0;str1[i]!='\0';i++)
    {
        for(j=0;str2[j]!='\0';j++)
        {
            if(str1[i]>str2[j])
            {
                printf("String 1 is greater than string 2\n");
                return ((int)str1[i]-(int)str2[j]);
            }
            else if(str2[j]>str1[i])
            {
                printf("String 2 is greater than string 1\n");
                return ((int)str2[j]-(int)str1[i]);
            }
        }
    }
}
void main()
{
    char str1[100],str2[100];
    int s;
    printf("Enter string 1");
    gets(str1);
    printf("Enter string 2");
    gets(str2);
    for(int i=0;str1[i]!='\0';i++)
    {
        str1[i]=tolower(str1[i]);
    }
    for(int j=0;str1[j]!='\0';j++)
    {
        str2[j]=tolower(str2[j]);
    }
    s=strcmpi(str1,str2);
    printf("The difference is %d",s);
}
```

Output:

```
D:\01-College\c programs\strings>gcc strcomp1.c -o strcomp1

D:\01-College\c programs\strings>strcomp1

Enter string 1: hello

Enter string 2: programming

String 2 is greater than string 1

The difference is 8
```

2) **Write a program to search the first occurrence of a substring in a given string without using library function.**

Code:

```
#include<stdio.h>
int occ(char str1[],char sub[])
{
    int i,j,l=0,m=0,f,pos;
    for(i=0;str1[i]!='\0';i++)
    {
        l++;
    }
    for(j=0;sub[j]!='\0';j++)
    {
        m++;
    }
    for(i=0;i<l;i++)
    {
        if(str1[i]==sub[0])
        {
            f=1;
            pos=i;
            for(j=0;j<m;j++)
            {
                if(str1[i]==sub[j])
                {
                    i++;
                }
                else
                {
                    f=0;
                    break;
                }

            }
            if(f==1)
            {
                return pos+1;
            }
        }
    }
    return pos+1;
}
```

```
void main()
{
    char str1[100],sub[50];
    int pos;
    printf("Enter main string: ");
    gets(str1);
    printf("Enter substring: ");
    gets(sub);
    pos=occ(str1,sub);
    if(pos!=0)
    {
        printf("The first occurence of the substring %s in the mainstring %s
is %d",sub,str1,pos);
    }
    else
    {
        printf("Substring is not found ");
    }

}
```

Output:

```
D:\01-College\c programs\strings>gcc strocc.c -o strocc
D:\01-College\c programs\strings>strocc
Enter main string: helloworldhellhe
Enter substring: ell
```

```
The first occurence of the substring ell in the mainstring
helloworldhellhe is 2
```

3) **Write a program to reverse a string without using the library function. No extra string should be used and the source string itself should be modified to store the reversed string. Number of exchanges should be minimum.**

Code:

```
#include<stdio.h>
void strrev(char str[])
{
    int l=0,i,j;
    char temp;
    for(int i=0;str[i]!='\0';i++)
    {
        l++;
    }
    for(i=0,j=l-1;i<=j/2;i++,j--)
    {
        temp=str[i];
        str[i]=str[j];
        str[j]=temp;
    }
}
void main()
{
    char str[100];
    printf("Enter string: ");
    gets(str);
    strrev(str);
    printf("The reversed string is %s",str);
}
```

Output:

```
D:\01-College\c programs\strings>gcc strrev.c -o strrev
D:\01-College\c programs\strings>strrev
Enter string: hello
The reversed string is olleh
```

4) Write an interactive C program that will encode or decode a line of text. To encode a line of text, proceed as follows.

a) Convert each character, including blank spaces, to its ASCII equivalent.

b) Generate a positive random integer. Add this integer to the ASCII equivalent of each character. The same random integer will be used for the entire line of text. After adding make sure the integer falls within ASCII range, so that the encoded character is always an ASCII character.

c) Display the characters that correspond to the encoded ASCII values. The procedure is reversed when decoding the line of text and also print the decoded text.

Code:

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

```
void main()
{

  char a[50],b[50]="";
  int x=0,n1=33,n2=126,y[50];
  Long int ascii=0,no,ran=0;
  printf("enter a string: ");
  gets(a);
  ran=rand();
  printf("The random number is: %d\n",ran);
  for (x=0;x<strlen(a);x++)
  {
     ascii=(int)a[x];
     y[x]=((ran+ascii)/n2)*n2;
     no=(ran+ascii)-y[x]+n1;
     b[x]=(char)no;
  }
  printf("encoded word is: %s\n",b);
  printf("\n");

  for (x=0;x<strlen(b);x++)
  {
    ascii=(int)b[x];
    no=ascii-ran-n1+y[x];
    a[x]=(char)no;
  }
  printf("\ndecoded word is: %s\n",a);
  printf("\n");
}
```

Output:

```
D:\01-College\c programs\strings>gcc strencdec.c -o strencdec
D:\01-College\c programs\strings>strencdec
enter a string: helloworld
The random number is: 41
encoded word is: 4188;C;>80
decoded word is: helloworld
```

Exercise 6

1) **Write an user-defined function in C that searches a given word in a line of text and returns the frequency count. Make use of pointer notation.**

Code:

```
#include<stdio.h>
#include<string.h>
int frequency(char* a,char* s)
{
    int t = 0,f = 0;
    char *str, *search;
    do
    {
       for (str = a, search = s ; *search != '\0' && *str ==*search ; search+
+,str++)
       {
       }
        if ( *search == '\0' )
        {
            f++;
            t = 1;
        }
    }while (*(a++));
    if (t)
    {
        return f;
    }
    return -1;
}
void main()
{
    char a[100],s[50];
    printf("\nEnter string : ");
    gets(a);
    printf("\nEnter word to search : ");
    gets(s);
    int f=frequency(a,s);
    printf("\nFrequency of given word : %d",f);
    printf("\n");
}
```

Output:

```
D:\01-College\c programs\pointers>gcc freqcount.c -o frqcount

D:\01-College\c programs\pointers>frqcount
```

```
Enter string : hello world hell hello

Enter word to search : hello

Frequency of given word : 2
```

2) Tokenising a line of text

Code:

```
#include<stdio.h>

#include<stdlib.h>
#include<string.h>
void main()
{
    int i=0,j=0,k=0,x=0,y=0,count=0;
    char string[100];
    puts("Enter the text ending with END : ");
    gets(string);
    for(i=0;string[i]!='\0';i++)
    {
        if(string[i]==' '||string[i]=='.')
        ++count;
    }
    char *words=(char)malloc((count+1)*sizeof(char));
    for(i=0;i<count;i++)
    {
        y=0;
        for(;string[x]!=' '&&string[x]!='.';x++,y++);
        words[j]=(char*)malloc(y*sizeof(char));
        x-=y;

        for(;string[x]!=' '&&string[x]!='.';x++)
        {
            *((words+j)+k)=string[x];
            k++;
        }
        (*words+j)[k]='\0';
        j++;
        k=0;
        x++;
    }
    words[i]=(char*)malloc(3*sizeof(char));
    strcpy(words[i],"END");
    count++;
    printf("Individual words : \n");
    for(i=0;i<count;i++)
    {
        printf("%s\n",words[i]);
        free(words[i]);
    }
    free(words);
}
```

Output:

```
Enter the text ending with END :
This is a C program using pointers. END
Individual words :
This
is
a
C
program
using
pointers
END
```

3)

Code:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int row,col,*a[10],*b[10],*c[10];
    printf("Enter number of rows: ");
    scanf("%d",&row);
    printf("Enter number of columns: ");
    scanf("%d",&col);
    for(int i=0;i<=row;i++)
    {
        a[i]=(int *)malloc(col*sizeof(int));
        b[i]=(int *)malloc(col*sizeof(int));
        c[i]=(int *)malloc(col*sizeof(int));
    }
    printf("Table A\n");
    for(int i=0;i<row;i++)
    {
        for(int j=0;j<col;j++)
        {
            printf("Enter data for row %d and column %d: ",i+1,j+1);
            scanf("%d",(a[i]+j));
        }
    }
    printf("Table B\n");
    for(int i=0;i<row;i++)
    {
        for(int j=0;j<col;j++)
        {
            printf("Enter data for r %d and column %d: ",i+1,j+1);
            scanf("%d",(b[i]+j));
        }
    }
    for(int i=0;i<row;i++)
    {
        for(int j=0;j<col;j++)
        {
            if(*(a[i]+j)>*(b[i]+j))
            {
                *(c[i]+j)=*(a[i]+j);
            }
            else
            {
                *(c[i]+j)=*(b[i]+j);
            }

        }
    }
    for(int i=0;i<row;i++)
    {
        for(int j=0;j<col;j++)
        {
            printf("%d\t",*(c[i]+j));
        }
        printf("\n");
    }

}
```

```
D:\01-College\c programs\pointers>gcc tables.c -o tables

D:\01-College\c programs\pointers>tables

Enter number of rows: 3

Enter number of columns: 3

Table A

Enter data for row 1 and column 1: 1

Enter data for row 1 and column 2: 2

Enter data for row 1 and column 3: 3

Enter data for row 2 and column 1: 4

Enter data for row 2 and column 2: 5

Enter data for row 2 and column 3: 6

Enter data for row 3 and column 1: 7
Enter data for row 3 and column 2: 8

Enter data for row 3 and column 3: 9

Table B

Enter data for row 1 and column 1: 9

Enter data for row 1 and column 2: 8

Enter data for row 1 and column 3: 7

Enter data for row 2 and column 1: 6

Enter data for row 2 and column 2: 5

Enter data for row 2 and column 3: 4

Enter data for row 3 and column 1: 3

Enter data for row 3 and column 2: 2

Enter data for row 3 and column 3: 1

Table C

9       8       7

6       5       6

7       8       9
```

4)

Code:

```
#include<stdio.h>
#include<stdlib.h>
void multiply(int **a,int **b,int r1,int c2,int c1)
{
    int c[r1][c2];
    for(int i=0;i<r1;i++)
    {
        for(int j=0;j<c2;j++)
        {
            c[i][j]=0;
            for(int k=0;k<c1;k++)
            {
                c[i][j]+=(*(*(a+i)+k) * *(*(b+k)+j));
            }
        }
    }
    printf("Elements of C : \n");
    for(int i=0;i<r1;i++)
    {
        for(int j=0 ;j<c2;j++)
        {
            printf("%d ",c[i][j]);
        }
        printf("\n");
    }
 free(c);
}
void main()
{
    int r1,c1,r2,c2;
    printf("Enter the row and column for 1st matrix: ");
    scanf("%d%d",&r1,&c1);
    printf("Enter the row and column for 2nd matrix: ");
    scanf("%d%d",&r2,&c2);
    int
    **a=(int**)malloc(sizeof(int)*r1),**b=(int**)malloc(sizeof(int)*r2);
    for(int i=0;i<r1;i++)
    {
        a[i]=(int*)malloc(sizeof(int)*c1);
    }
    for(int i=0;i<r2;i++)
    {
        b[i]=(int*)malloc(sizeof(int)*c2);
    }




    printf("Enter the elements of A : ");
    for(int i=0;i<r1;i++)
    {
        for(int j=0;j<c1;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the elements of B : \n");
    for(int i=0;i<r2;i++)
    {
        for(int j=0;j<c2;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    if(c1==r2)
    {
        void (*p)(int**,int**,int,int,int)=multiply;
        (*p)(a,b,r1,c2,c1);
    }
    free(a);
    free(b);
}
```

Output:

```
D:\01-College\c programs\pointers>multiply

Enter the row and column for 1st matrix: 3 3

Enter the row and column for 2nd matrix: 3 2

Enter the elements of A : 1

2

3

4

5

6

7

8

9




Enter the elements of B :

1

2

3

4
```

```
5

6

Elements of C :

22 28

49 64

76 100
```

Exercise 7- Structures

1)Generating salary slip of a single employer

Code:

```
#include<stdio.h>
#include<string.h>
struct employer
{
    int id;
    char emp_name[50];
    char designation[10];
    float bp;
    float da;
    float hra;
    float cca;
    float gp;
    float ded;
    float np;
}emp,*ptr=&emp;


void input(struct employer *ptr)
{

    printf("Enter employer ID: ");
    scanf("%d",&ptr->id);
    printf("Enter employer name: ");
    scanf("%s",ptr->emp_name);
    printf("Enter designation: ");
    scanf("%s",ptr->designation);
    printf("Enter basic pay: ");
    scanf("%f",&ptr->bp);
}


void calculate(struct employer *ptr)
{
    printf("The salary slip of the employer is generated\n");
    ptr->da=0.88*(ptr->bp);
    ptr->hra=0.08*(ptr->bp);
    ptr->cca=1000;
    ptr->gp=ptr->bp+ptr->da+ptr->hra+ptr->cca;
    ptr->ded=1000+0.1*(ptr->bp);
    ptr->np=ptr->gp-ptr->ded;
    return;
}
```

```
void display(struct employer *ptr)
{
    printf("Employer ID: %d\n",ptr->id);
    printf("Employer name: %s\n",ptr->emp_name);
    printf("Designation: %s\n",ptr->designation);
    printf("Basic pay: %f\n",ptr->bp);
    printf("HRA : %f\n",ptr->hra);
    printf("CCA: %f\n",ptr->cca);
    printf("Gross pay: %f\n",ptr->gp);
    printf("Deductions: %f\n",ptr->ded);
    printf("Net pay: %f\n",ptr->np);
    printf("\n");
    return;
}

void main()
{
    struct employer *ptr;
    input(&ptr);
    calculate(&ptr);
    display(&ptr);
}
```

Output:

```
D:\01-College\c programs\structures>gcc payslip.c -o payslip
D:\01-College\c programs\structures>payslip

Enter employer ID: 1324
Enter employer name: Raju
Enter designation: Manager
Enter basic pay: 12000
The salary slip of the employer is generated
Employer ID: 1324
Employer name: Raju
Designation: Manager
Basic pay: 12000.000000
HRA : 960.000000
CCA: 1000.000000
Gross pay: 24520.000000
Deductions: 2200.000000
Net pay: 22320.000000
```

2)Generating marksheet of n students

Code:

```
#include<stdio.h>
#include<string.h>
typedef struct
{
    int d;
    int m;
    int y;
}date;
date dob;


enum gender
{
    Male=1,Female=2,Trangender=3
}gen;


typedef struct
{
    long int rollnum;
    char name[20];
    int marks[5];
    int totalmarks;
    int rank;
    date dob;
    int age;
    enum gender gen[15];
}student;
student stud[20];
void rank(student stud[20],int n);

void main()
{
    student stud[20];
    int n,i,j;
    printf("Enter number of students: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        stud[i].totalmarks=0;
        printf("Student number %d\n",i+1);
        printf("Enter roll number: ");
        scanf("%d",&stud[i].rollnum);
        printf("Enter name: ");
        scanf("%s",stud[i].name);
        for(j=0;j<5;j++)
        {
            printf("Enter mark %d: ",j+1);
            scanf("%d",&stud[i].marks[j]);
            stud[i].totalmarks+=stud[i].marks[j];
        }
        printf("Enter the date in dd/mm/yyyy format: ");
        scanf("%d %d %d",&stud[i].dob.d,&stud[i].dob.m,&stud[i].dob.y);
        printf("Enter age: ");
        scanf("%d",&stud[i].age);
        printf("\n");
    }
    printf("\n");
    rank(stud,n);
    display(stud,n);
}
void rank(student stud[20],int n)
{
    student k;
    for(int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            k=stud[i];
            stud[i]=stud[j];
            stud[j]=k;
        }
    }
    for(int i=0;i<n;i++)
    {
        stud[i].rank=i+1;
    }
}
void display(student stud[20],int n)
{
    printf("\t\tMarksheet\n");
    printf("\n");
    for(int i=0;i<n;i++)
    {
        printf("Roll number: \t%ld\n",stud[i].rollnum);
        printf("Student name: \t%s\n",stud[i].name);
        printf("Date of birth: \t%d/%d/%d\n",stud[i].dob.d,stud[i].dob.m,stud
[i].dob.y);
        printf("Age: \t%d\n",stud[i].age);
        printf("Marks of the student\n");
        for(int j=0;j<5;j++)
        {
            printf("Mark %d: \t%d\n",j+1,stud[i].marks[j]);
        }
        printf("Total marks: \t%d\n",stud[i].totalmarks);
        printf("Rank: \t%d\n",stud[i].rank);
        printf("\n");
    }
}
```

Output:

```
Enter number of students: 5
Student number 1
Enter roll number: 1876
Enter name: Shreya
Enter mark 1: 98
Enter mark 2: 99
Enter mark 3: 98
Enter mark 4: 97
Enter mark 5: 90
Enter the date in dd/mm/yyyy format: 12 10 2001
Enter age: 18

Student number 2
Enter roll number: 2987
Enter name: Sam
Enter mark 1: 98
Enter mark 2: 87
Enter mark 3: 100
Enter mark 4: 100
Enter mark 5: 99
Enter the date in dd/mm/yyyy format: 10 10 2001
Enter age: 18

Student number 3
Enter roll number: 1873
Enter name: Sarah
Enter mark 1: 98
Enter mark 2: 97
Enter mark 3: 99
Enter mark 4: 99
Enter mark 5: 91
Enter the date in dd/mm/yyyy format: 13 10 2001
Enter age: 18

Student number 4
Enter roll number: 1765
Enter name: Ram
Enter mark 1: 99
Enter mark 2: 99
Enter mark 3: 99
Enter mark 4: 99
Enter mark 5: 100
Enter the date in dd/mm/yyyy format: 14 2 2002
Enter age: 17

Student number 5
Enter roll number: 1987
Enter name: Sohan
Enter mark 1: 98
Enter mark 2: 98
Enter mark 3: 99
Enter mark 4: 90
Enter mark 5: 97
Enter the date in dd/mm/yyyy format: 13 4 2001
Enter age: 18


                Marksheet

Roll number:    1987
Student name:   Sohan
Date of birth:  13/4/2001
Age:    18
Marks of the student
Mark 1:         98
Mark 2:         98
Mark 3:         99
Mark 4:         90
Mark 5:         97
Total marks:    482
Rank:   1

Roll number:    1765
Student name:   Ram
Date of birth:  14/2/2002
Age:    17
Marks of the student
Mark 1:         99
Mark 2:         99
Mark 3:         99
Mark 4:         99
Mark 5:         100
Total marks:    496
Rank:   2

Roll number:    1873
Student name:   Sarah
Date of birth:  13/10/2001
Age:    18
Marks of the student
Mark 1:         98
Mark 2:         97
Mark 3:         99
Mark 4:         99
Mark 5:         91
Total marks:    484
Rank:   3

Roll number:    2987
Student name:   Sam
Date of birth:  10/10/2001
Age:    18
Marks of the student
Mark 1:         98
Mark 2:         87
Mark 3:         100
Mark 4:         100
Mark 5:         99
Total marks:    484
Rank:   4

Roll number:    1876
Student name:   Shreya
Date of birth:  12/10/2001
Age:    18
Marks of the student
Mark 1:         98
Mark 2:         99
Mark 3:         98
Mark 4:         97
Mark 5:         90
Total marks:    482
Rank:   5
```

Exercise 8- File handling

```
 #include<stdio.h>
#include<stdlib.h>
void main(int argc, char* argv[])
{
char ch,x;
FILE *f1,*f2;
f1=fopen(argv[1],"r");
if(f1==NULL)
{
printf("\nFile does not exist");
}
f2=fopen(argv[2],"r");
if(f2==NULL)
{
fclose(f2);
f2=fopen(argv[2],"w");
while((x=getc(f1))!=EOF)
{putc(x,f2);}
fclose(f1);
fclose(f2);
}
else
{
printf("\nEnter w to overwrite or a to append : ");
scanf("%c",&ch);
if(ch=='w')
{
fclose(f2);
f2=fopen(argv[2],"w");
while((x=getc(f1))!=EOF){ putc(x,f2);}
fclose(f1); fclose(f2);
}
else if(ch=='a')
{
fclose(f2);
f2=fopen(argv[2],"a");
while((x=getc(f1))!=EOF)
{
putc(x,f2);}
fclose(f1);
fclose(f2);
}
}
printf("\nContents of f2 after copying : \n");
f2=fopen(argv[2],"r");
while((x=getc(f2))!=EOF)
{
printf("%c",x); }}
```

Output:

```
Command line arguments : 2 s.txt d.txt
Contents of s.txt : Programming in C.
Contents of d.txt : Hello world.
Enter w to overwrite or a to append : a
Contents of f2 after copying :
Hello World. Programming in C.
```

2) **Code:**

```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
struct rec
{
char name[20];
char addr[25];
long int no;
}r[20],record;
void main()
{
char s[20];
FILE *f1;
f1=fopen("file.txt","w");
int n;
printf("\nEnter no. of records : ");
scanf("%d",&n);
printf("\nEnter the details : \n");
for(int i=0;i<n;i++)
{ printf(“Enter name”);
scanf("%s",r[i].name);
printf(“Enter address”);
scanf("%s",r[i].addr);
printf(“Enter Number”);
scanf("%ld",&r[i].no);
fprintf(f1," %s %s %ld",r[i].name,r[i].addr,r[i].no);
}
fclose(f1);
printf("\nEnter record to be appended : \n");
printf(“Enter name”);
scanf("%s",record.name);
printf("Enter address");
scanf("%s",record.addr);
printf("Enter Number");
scanf(" %ld",&record.no);
f1=fopen("file.txt","a");
fprintf(f1," %s %s %ld",record.name,record.addr,record.no);
n++;
fclose(f1);
printf("\nEnter name of record to search : ");
scanf("%s",s);
f1=fopen("file.txt","r");
for(int i=0;i<n;i++)
{
fscanf(f1," %s %s %ld",&record.name,&record.addr,&record.no);
if(strcmp(record.name,s)==0)
{
printf(" %s %s %ld",record.name,record.addr,record.no);
}
}
fclose(f1);
f1=fopen("file.txt","r");
printf("\nDetails of all records : ");
for(int i=0;i<n;i++)
{
fscanf(f1," %s %s %ld",&record.name,&record.addr,&record.no);
printf("\n %s %s %ld",record.name,record.addr,record.no);
}
fclose(f1);
}
```

Output:

```
Enter no. of records : 2
Enter the details :
Sam
TNagar
12345
Ram
BesantNagar
23456
Enter record to be appended :
Sid
Nungambakkam
34567
Enter name of record to search : Ram
Ram
```

```
BesantNagar
23456
Details of all records :
Sam
TNagar
12345
Ram
BesantNagar
23456
Sid
Nungambakkam
34567
```

3) Code:

```
#include<stdio.h>
#include<string.h>
struct data
{
char name[20];
char address[20];
long int telephone;
};
void display(struct data s[100],int a);
void adddata(struct data s[100],int a)
{ FILE *fp,*fp1;
fp=fopen("data.bin","rb+");
struct data temp;
int n;
printf("\enter location to be inserted");
scanf("%d",&n);
printf("\n enter name ");
scanf("%s",temp.name);
printf("\n enter address ");
scanf("%s",temp.address);
printf("\n enter telephone number ");
scanf("%ld",&(temp.telephone));
a+=1;
fp1=fopen("sample.bin","wb");
for(int i=a-1;i>=n-1;i--)
{
s[i+1]=s[i];
}
s[n-1]=temp;
```

```
for(int i=n;i<a-1;i++)
{
fread(s+i,sizeof(struct data),1,fp);
fwrite(s+i,sizeof(struct data),1,fp1);
}
rewind(fp1);
fseek(fp,(n-1)*sizeof(struct data),SEEK_SET);
fwrite(s+n-1,sizeof(struct data),1,fp);
for(int i=n;i<a;i++)
{
fread(s+i,sizeof(struct data),1,fp1);
fwrite(s+i,sizeof(struct data),1,fp);
}
fclose(fp);
fclose(fp1);
}
void delete(struct data s[100],int a)
{
int n;
FILE *fp,*fp1;
printf("enter item to be deleted");
scanf("%d",&n);
for(int i=n-1;i<a;i++)
{
s[i]=s[i+1];
}
a--;
fp=fopen("data.bin","wb");
fp1=fopen("sample.bin","wb");
for(int i=0;i<a;i++)
{
fwrite(s+i,sizeof(struct data),1,fp1);
}
display(s,a);
remove("data.bin");
rename("sample.bin","data.bin");
}
void search(struct data s[100],int a)
{ FILE *fp,*fp1;
fp=fopen("data.bin","rb");
int n;
printf("enter item number");
scanf("%d",&n);
fseek(fp,(n-1)*sizeof(struct data),0);
if(fread(s+n-1,sizeof(struct data),1,fp))
{ printf("%s\n%s\n%ld\n",s[n-1].name,s[n-1].address,s[n-1].telephone);
}
fclose(fp);
display(s,a);
```

```
}
void display(struct data s[100],int a)
{ FILE *fp;
fp=fopen("data.bin","rb");
for(int i=0;i<a;i++)
{
if(fread(s+i,sizeof(struct data),1,fp)!=EOF)
{
printf("%s\n%s\n%ld\n",s[i].name,s[i].address,s[i].telephone);
}
}
fclose(fp);
}
void main()
{ int n;
printf("\n enter number of data sets");
scanf("\n %d",&n);
struct data s[100];
FILE *fp;
fp=fopen("data.bin","wb");
for(int i=0;i<n;i++)
{
printf("enter name\n");
scanf("%s",s[i].name);
printf("enter address");
scanf("%s",s[i].address);
printf("enter phone number");
scanf("%ld",&(s[i].telephone));
fwrite(s+i,sizeof(struct data),1,fp);
}
fclose(fp);
int opn;
display(s,n);
printf("enter opn, 1 to add, 2 to delete,3 to search,any other number to
exit");
scanf("%d",&opn);
switch(opn)
{
case 1: adddata(s,n);
n++;
display(s,n);
break;
case 2: delete(s,n);
n--;
break;
case 3: search(s,n);
break; }
}
```

Output:

```
Enter no. of records : 3
Enter the details :
Sam
TNagar
12345
Ram
BesantNagar
23456
Sid
Nungambakkam
34567
Enter name of record to delete : Ram
Sam
TNagar
12345
Sid
Nungambakkam
34567
Enter position of record to display : 1
Sam
TNagar
12345
```