

1. PRACTISE SESSION

1. a) Write a simple program that prints the results of all the operators available in C (including pre/ post increment, bitwise and/or/not , etc.). Read required operand values from standard input.

AIM : To perform operations using all operators

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,result,result2;
float m;
clrscr();
printf("enter a and b values\n");
scanf("%d%d",&a,&b);
printf("addition of a and b is %d\n",a+b);
printf("substraction of a and b is %d\n",a-b);
printf("multiplication of a and b is %d\n",a*b);
printf("remainder of a and b is %d\n",a%b);
printf("division of a and b is %d\n",a/b);
result=(a>0)&&(b<=10);
printf("Logical And=%d\n",result);
result2=(a==b)||!(b!=0);
printf("Logical OR NOT=%d\n",result2);
printf("Bitwise And %d\n",a&b);
printf("Bitwise Or %d\n",a|b);
printf("Bitwise NOT %d\n",!a);
printf("Bitwise complement %d\n",~a);
printf("Bitwise XOR %d\n",a^b);
printf("Bitwise Shift right %d\n",a>>2);
printf("Bitwise Shift left %d\n",b<<3);
printf("integer size=%d,floating point size=%d\n",sizeof(a),sizeof(m));
printf("conditional expression=%d\n",(a-b>50)?100:200);
printf("preincrement %d\n",++a);
printf("postincrement %d\n",b++);
printf("predecrement %d\n",--a);
printf("postdecrement %d\n",b--);
getch();
}
```

}

INPUT: enter a and b values

20 30

OUTPUT:

addition of a and b is 50

subtraction of a and b is -10

multiplication of a and b is 600

remainder of a and b is 2

division of a and b is 0

Logical And=

Logical OR NOT

Bitwise And

Bitwise Or

Bitwise NOT

Bitwise complement

Bitwise XOR

Bitwise Shift right

Bitwise Shift left

- b) Write a simple program that converts one given data type to another using auto conversion and casting. Take the values form standard input.

AIM: To convert given data type to another type using type casting.

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,x;
    float f;
    double p;
    short s;
    clrscr();
    printf("enter integer value\n");
    scanf("%d",&i);
    printf("enter float value\n");
    scanf("%f",&f);
    p=i; // implicit conversion
    printf("implicit value is %lf\n",p);
    x= (int) f; // Explicit conversion
    printf("Explicit value is %d\n",x);
    getch();
}
```

INPUT:

enter integer value

2

enter float value

18.65

implicit value is 2.0000

Explicit value is 18

2. SIMPLE NUMERIC PROBLEMS

2 a) Write a program for find the max and min from the three numbers.

AIM: To find maximum and minimum of three numbers

Source code:

```
#include<stdio.h>

#include<conio.h>
void main()
{
    int a,b,c;
    printf("Enter 3 numbers");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b && a>c)
        printf("Maximum number is a = %d",a);
    else if(b>a && b>c)
        printf("Maximum number is b = %d",b);
    else
        printf("Maximum number is c = %d",c);
    if(a<b && a<c)
        printf("Minimum number is a = %d",a);
    else if(b<a && b<c)
        printf("Minimum number is b = %d",b);
    else
        printf("Minimum number is c = %d",c);
}
```

INPUT: Enter 3 numbers

10 20 5

OUTPUT:

Maximum number b=20

Minimum number c=5

b) Write the program for the simple compound interest.

AIM: To compute simple compound interest.

Source code:

```
#include<stdio.h>
#include<math.h>
int main()
{
    float p,q,r,SI,CI;
    int n;
    printf("Enter the value of Principal p = ");
    scanf("%f",&p);
    printf("Enter the value of Rate r = ");
    scanf("%f",&r);
    printf("Enter the value of Period in year n = ");
    scanf("%d",&n);
    SI = ((p*r*n)/100);
    printf("Simple Interest SI=%f \n",SI);
    q = 1+(r/100);
    CI=p*pow(q,n)-p;
    printf("Compound Interest CI=%f \n",CI);
    return 0;
}
```

INPUT:

Enter the value of Principal p =1000

Enter the value of Rate r =12

Enter the value of Period in year n =2

OUTPUT:

Simple Interest SI=240.000000

Compound Interest CI=254.400009

c) Write program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >=70% = Distinction. Read percentage from standard input.

AIM: To calculate the percentage of students marks.

Source Code:

```
#include <stdio.h>
int main()
{
    int phy, chem, bio, math, comp;
    float per;
    /* Input marks of five subjects from user */
    printf("Enter five subjects marks: ");
    scanf("%d%d%d%d%d", &phy, &chem, &bio, &math, &comp);
    /* Calculate percentage */
    per = (phy + chem + bio + math + comp) / 5.0;
    printf("Percentage = %.2f\n", per);
    /* Find grade according to the percentage */
    if(per >= 70)
    {
        printf("Distinction");
    }
    else if(per >= 60||per<=70)
    {
        printf("First Class");
    }
    else if(per >= 40|| per<=60)
    {
        printf("Second class");
    }

    else
    {
        printf("Failed");
    }

    return 0;
}
```

INPUT:

Enter five subjects marks:

95 95 97 98 90

OUTPUT:

Percentage= 95.00

Distinction

d) Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15

AIM: To print a multiplication table

Source Code:

```
#include <stdio.h>
int main()
{
    int n, i, rows;
    printf("Enter an integer: ");
    scanf("%d", &n);
    printf("enter the number of rows\n");
    scanf("%d", &rows);
    for(i=1; i<=rows; i++)
    {
        printf("%d * %d = %d \n", n, i, n*i);
    }
    return 0;
}
```

INPUT:

Enter an integer: 5
enter the number of rows: 3

OUTPUT:

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15

- e) Write a program that shows the binary equivalent of a given positive number between 0 to 255.

AIM: To convert given positive number into binary equivalent

Source Code:

```
#include <stdio.h>
#include <math.h>
long decimalToBinary(int decimalnum)
{
    long binarynum = 0;
    int rem, temp = 1;

    while (decimalnum!=0)
    {
        rem = decimalnum%2;
        decimalnum = decimalnum / 2;
        binarynum = binarynum + rem*temp;
        temp = temp * 10;
    }
    return binarynum;
}

int main()
{
    int decimalnum;
    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);
    printf("Equivalent Binary Number is: %ld",
        decimalToBinary(decimalnum));
    return 0;
}
```

INPUT:

Enter a Decimal Number: 234

OUTPUT:

Equivalent Binary Number is: 11101010

3. EXPRESSION EVALUATION

- a) A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula $s = ut + (1/2)at^2$ where u and a are the initial velocity in m/sec ($= 0$) and acceleration in m/sec^2 ($= 9.8 m/s^2$)).

AIM: To calculate the time.

Source Code:

```
#include <stdio.h>
#include <math.h>
void main()
{
    int tim_intrval, counter,time;
    float accl, distance=0, velos;
    clrscr();
    printf("<=====PROGRAM FOR CALC TOTAL DISTANCE
    TRAVELED BY A VECHIAL=====>");
    printf("\n\n\t\tNO OF TIME INTERVALS : ");
    scanf("%d",&tim_intrval);
    for(counter = 1; counter <= tim_intrval; counter++)
    {
        printf("\n\t\tAT T%d TIME(sec) : ",counter);
        scanf("%d",&time);
        printf("\t\tVELOCITY AT %d sec (m/sec) : ",time);
        scanf("%f",&velos);
        printf("\t\tACCLERATION AT %d sec (m/sec^2): ",time);
        scanf("%f",&accl);
        distance += (velos*time + (accl*pow(time,2))/2);
    }
    printf("\n\n\tTOTAL DISTANCE TRAVELLED BY VEHICLE IN %d
    INTERVALS OF TIME : %f",tim_intrval,distance);
    getch();
}
```

Output:

```
NO OF TIME INTERVALS :2
AT T1 TIME(sec) :5
VELOCITY AT 5 sec (m/sec) :20
ACCLERATION AT 5 sec (m/sec^2):35
AT T1 TIME(sec) :10
VELOCITY AT 10 sec (m/sec) :30
ACCLERATION AT 10 sec (m/sec^2):35
TOTAL DISTANCE TRAVELLED BY VEHICLE IN 2 INTERVALS OF
TIME :2587.5
```

- b) Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use Switch Statement)

AIM: To perform all arithmetic operations

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
main()
{
    int a, b, c;
    char op;
    //clrscr();
    printf("Enter a and b:");
    scanf("%d%d",&a,&b);
    do
    {
        printf("\n\nMENU\n");
        printf("+ Addition\n");
        printf("- Subtraction\n");
        printf("* Multiplication\n");
        printf("/ Division\n");
        printf("%%% Remainder\n");
        printf("E Exit\n");
        printf("Enter your choice :");
        getchar();
        op=getchar();
        if(op=='E'||op=='e')
            exit(1);
        switch(op)
        {
            case '+':
                printf("Addition\n");
                c=a+b;
                printf("Sum=%d\n",c);
                break;
            case '-':
                printf("Subtraction\n");
                c=a-b;
                printf("Difference=%d\n",c);
                break;
            case '*':
                printf("Multiplication\n");
                c=a*b;
                printf("Product=%d\n",c);
```

```
        break;

    case '/':
        printf("Division\n");
        c=a/b;
        printf("Quotient=%d\n",c);
        break;
    case '%':
        printf("Remainder\n");
        c=a%b;
        printf("Remainder=%d\n",c);
        break;
    default:
        printf("Invalid Option\n");
        break;
} /*end of switch statement*/
}
while(1); /*End of while*/

}/*End of main function*/
```

INPUT:

Enter a and b:20 10

MENU

+ Addition

- Subtraction

* Multiplication

/ Division

% Remainder

E Exit

Enter your choice : +

OUTPUT:

Sum= 30

c) Write a program that finds if a given number is a prime number

AIM: To check given number is prime or not

Source Code:

```
#include <stdio.h>
int main()
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    for(i=2; i<=n/2; ++i)
    {
        // condition for nonprime number
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }

    if (flag==0)
        printf("%d is a prime number.",n);
    else
        printf("%d is not a prime number.",n);

    return 0;
}
```

INPUT:

Enter a positive integer:

11

OUTPUT:

11 is a prime number

- d) Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.

AIM: To find the sum of individual digits and check for palindrome

Source Code:

```
#include <stdio.h>
void main()
{
    long num, temp, digit, sum = 0, reverse = 0;
    printf("Enter the number \n");
    scanf("%ld", &num);
    temp = num;
    while (num > 0)
    {
        digit = num % 10;
        reverse = reverse * 10 + digit;
        sum = sum + digit;
        num /= 10;
    }
    printf("Given number = %ld\n", temp);
    printf("Sum of the digits %ld = %ld\n", temp, sum);
    if(reverse == temp)
        printf("Given number is palindrome = %ld\n", temp);
    else
        printf("Given number is not a palindrome = %ld\n", temp);
}
```

INPUT:

Enter the number 121

OUTPUT:

Given number =121

Sum of the digits=4

Given number is palindrome =121

- e) A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

AIM: To generate Fibonacci series

Source Code:

```
#include <stdio.h>
void main()
{
    int num1=0, num2=1,no,counter,fab;
    clrscr();
    printf("<-----PROGRAM TO FIND THE
FIBONACCI SERIES UP TO N NO. IN SERIES----->");
    printf("\n\n\n\tENTER LENGTH OF SERIES (N) : ");
    scanf("%d",&no);
    printf("\n\n\t<---FIBONACCI SERIES--->");
    printf("\n\n\t%d %d",num1,num2);
    for(counter = 1; counter <= no-2; counter++)
    {
        fab=num1 + num2;
        printf(" %d",fab);
        num1=num2;
        num2=fab;
    }
    getch();
}
```

INPUT:

ENTER THE LENGTH OF THE SERIES N 10

OUTPUT:

FIBONACCI SEQUENCE FOR THE FIRST 10 TERMS:

0 1 1 2 3 5 8 13 21 34

- f) Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

AIM: To generate prime number between 1 to n

Source Code

```
#include<stdio.h>
#include<conio.h>void main()
{
    int n,i,fact,j;
    clrscr();

    printf("enter the number:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {

        fact=0;
        for(j=1;j<=i;j++)
        {
            if(i%j==0)
                fact++;
        }
        if(fact==2)
            printf("\n%d",i);
        }
        getch( );
    }
```

INPUT:

Enter the number 10

OUTPUT:

3 5 7 are prime numbers

g) Write a C program to find the roots of a Quadratic equation.

AIM: To find roots of quadratic equation.

Source Code

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int a,b,c;
    float disc,root1,root2;
    float img,real;
    clrscr();
    printf("enter values for a,b,c\n");
    scanf("%d%d%d",&a,&b,&c);
    disc=(float)b*b-4*a*c;
    if(disc>0)
    {
        printf("the roots are real&unequal\n");
        root1=(-b+sqrt(disc))/(2*a);
        root2=(-b-sqrt(disc))/(2*a);
        printf("root1=%f\n",root1);
        printf("root2=%f\n",root2);
    }
    else if(disc==0)
    {
        printf("the roots are real and equal\n");
        root1=-b/(2*a);
        root2=root1;
        printf("root1=%f\n",root1);
        printf("root2=%f\n",root2);
    }
    else
    {
        printf("the roots are imaginary\n");
        disc=-disc;
        img=(float)disc/2*a;
        real=(float)-b/2*a;
        if(img>0)
        {
            printf("root1=%f+i%f\n",real,img);
            printf("root2=%f-i%f\n",real,img);
        }
        else
        {

```



```
img=-img;
printf("root1=%f+i%f\n",real,img);

printf("root2=%f-i%f\n",real,img);
}
}
getch();
}
```

INPUT:

ENTER VALUESFOR a b c

1 4 4

OUTPUT:

THE ROOTS ARE EQUAL AND THEY ARE..

Root1=-2

Root2=-2

- h) Write a C program to calculate the following, where x is a fractional value

$$1 - x/2! + x^2/4! - x^3/6! + \dots$$

AIM: To find roots of quadratic equation.

Source Code

```
#include <stdio.h>
#include <math.h>
void main()
{
    int counter,f_coun;
    float sum=0,x,power,fact;
    clrscr();
    printf("<-----PROGRAM FOR SUM OF EQ. SERIES----->");
    printf("\n\n\tEQUATION SERIES : 1- X^2/2! + X^4/4! - X^6/6! + X^8/8! - X^10/10!");
    printf("\n\n\tENTER VALUE OF X : ");
    scanf("%f",&x);
    for(counter=0, power=0; power<=10; counter++,power=power+2)
    {
        fact=1;
        /*CALC FACTORIAL OF POWER VALUE*/
        for(f_coun=power; f_coun>=1; f_coun--)
            fact *= f_coun;
        /*EQ. FOR SUM SERIES */
        sum=sum+(pow(-1,counter)*(pow(x,power)/fact));
    }
    printf("SUM : %f",sum);
    getch();
}
```

INPUT:

ENTER VALUE OF X

1

OUTPUT:

Sum 0.540302

- i) Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\dots+x^n$. For example: if n is 3 and x is 5, then the program computes $1+5+25+125$.

AIM: To generate geometric progression

Source code:

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int first, ratio, terms, value, sum=0, i;
    printf("Enter the number of terms in GP series\n");
    scanf("%d", &terms);
    printf("Enter first term and common ratio of GP series\n");
    scanf("%d %d", &first, &ratio);
    /* print the series and add all elements to sum */
    value = first;
    printf("GP SERIES\n");
    for(i = 0; i < terms; i++)
    {
        printf("%d ", value);
        sum += value;
        value = value * ratio;
    }
    printf("\nSum of the GP series till %d terms is %d\n", terms, sum);
    getch();
    return 0;
}
```

INPUT:

Enter the number of terms in GP series

6

Enter first term and common ratio of GP series

2 4

OUTPUT:

GP SERIES

2 4 8 16 32 64

Sum of the GP series till 6 terms is 126

4. ARRAYS AND POINTERS AND FUNCTIONS

- a. Write a C program to find the minimum, maximum and average in an array of integers.

AIM: To find max , min and average from an array.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,n,sum =0,min,max,list[100];
    float avg;
    clrscr();
    printf("Enter no of elements :");
    scanf("%d",&n);
    printf("\nEnter %d elements \n",n);
    for(i=0;i<n;i++)
        scanf("%d",&list[i]);
    min=max=sum= list[0];
    for(i=1;i<n;i++)
    {
        if(list[i]>max)
            max=list[i];
        if(list[i]<min)
            min=list[i];
        sum+=list[i];
    }
    avg=sum/(float)n;
    printf("The list of elements are:\n");
    for(i=0;i<n;i++)
        printf("%5d",list[i]);
    printf("\n\n");
    printf("Maximum Value  = %d\n",max);
    printf("Minimum Value  = %d\n",min);
    printf("Average        = %f\n",avg);
    getch();
}
```

INPUT:

Enter the elements in to the array: 5

Enter 5 elements:

2 6 1 5 7

OUTPUT:

The list of elements are:

2 6 1 5 7

Maximum Value = 7

Minimum Value = 1

Average = 4.20000

- a) Write a functions to compute mean, variance, Standard Deviation, sorting of n elements in single dimension array.

AIM: To compute mean, variance, Standard Deviation, sorting of n elements in single dimension array.

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define MAXSIZE 10
void main()
{
    float x[MAXSIZE];
    int i, n;
    float average, variance, std_deviation, sum = 0, sum1 = 0;
    clrscr();
    printf("Enter the value of N \n");
    scanf("%d", &n);
    printf("Enter %d real numbers \n", n);
    for (i = 0; i < n; i++)
    {
        scanf("%f", &x[i]);
    }
    for (i = 0; i < n; i++)
    {
        sum = sum + x[i];
    }
    average = sum / (float)n;
    for (i = 0; i < n; i++)
    {
        sum1 = sum1 + pow((x[i] - average), 2);
    }
    variance = sum1 / (float)n;
    std_deviation = sqrt(variance);
    printf("Average of all elements = %.2f\n", average);
    printf("variance of all elements = %.2f\n", variance);
    printf("Standard deviation = %.2f\n", std_deviation);
}
```

INPUT:**Enter the value of N****5****Enter 5 real numbers****34****88****32****12****10****OUTPUT:****Average of all elements = 35.20****variance of all elements = 794.56****Standard deviation = 28.19**

c) Write a C program that uses functions to perform the following:

AIM: To perform addition of matrices.

Source Code:

i) Addition of Two Matrices

```
#include<stdio.h>
#include<conio.h>
void add(int A[10][10],int B[10][10],int m,int n);
void main()
{
    int i,j;
    clrscr();
    printf("Enter the no. of rows\n");
    scanf("%d",&m);
    printf("Enter the no. of columns\n");
    scanf("%d",&n);
    printf("Enter the elements of matrix A\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&A[i][j]);
        }
    }
    printf("Enter the elements of matrix B\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&B[i][j]);
        }
    }
    add(A,B,m,n);
}
void add(int A[10][10],int B[10][10],int m,int n)
{
    int C[10][10],i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
```



```
{  
C[i][j]=A[i][j]+B[i][j];  
}  
}  
printf("The addition of two matrices A and B is\n");  
for(i=0;i<m;i++)  
{  
for(j=0;j<n;j++)  
{  
printf("%3d",C[i][j]);  
}  
printf("\n");  
}  
getch();  
}
```

INPUT:

Enter the no. of rows:3

Enter the no. of columns:3

Enter the elements of matrix A

1	2	3
4	5	6
7	8	9

Enter the elements of matrix B

1	1	1
1	1	1
1	1	1

OUTPUT:

The addition of two matrices A and B is

2	3	4
5	6	7
8	9	10

ii. Multiplication of Two Matrices**AIM: To multiply two matrices****Souce Code:**

```
#include<stdio.h>
#include<conio.h>
void multiply(int a[10][10],int b[10][10],int,int,int,int);
void main()
{
int a[10][10],b[10][10],r1,c1,r2,c2,i,j;
clrscr();
printf("Enter no of rows and columns of matrix A\n");
scanf("%d%d",&r1,&c1);
printf("Enter no of rows and columns of matrix B \n");
scanf("%d%d",&r2,&c2);
if(c1==r2)
printf("Matrix multiplication is possible");
else
exit();
printf("Enter the elements of matrix A\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("Enter the elements of matrix B\n");
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
{
scanf("%d",&b[i][j]);
}
}
multiply(a,b,r1,c1,r2,c2);
}
void multiply(int a[10][10], int b[10][10],int r1,int c1,int r2,int c2)
{
```

```
int i,j,k,c[10][10];
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
c[i][j]=0;
for(k=0;k<r2;k++)
{
c[i][j]=c[i][j]+a[i][k]*b[k][j];
}
}
}
printf("The resultant matrix multiplication is \n");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
printf("%3d",c[i][j]);
}
printf("\n");
}
getch();
}
```

INPUT:

Enter number of rows & columns of Matrix A:

2 2

Enter number of rows & columns of MatrixB:

2 2

Enter the elements of matrix A: 2 2 2 2

Enter the elements of matrix B: 2 2 2 2

OUTPUT:

The resultant matrix multiplication is:

8 8

8 8

iii. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be same.

AIM: To transpose a given matrix.

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
void trans(int,int,int[10][10]);
int i,j,a,b,m[10][10];
clrscr();
printf("enter the rows and columns of matrix");
scanf("%d%d",&a,&b);
printf("enter the elements");
for(i=1;i<=a;i++)
{
for(j=1;j<=b;j++)
{
printf("enter m[%d][%d]...",i,i);
scanf("%d",&m[i][j]);
}
}
printf("\n before transpose\");
for(i=1;i<=a;i++)
{
for(j=1;j<=b;j++)
{
printf("%d\t",m[i][j]);
}
printf("\n");
}
trans(a,b,m);
getch();
}
void trans(inta,intb,intm[10][10])
{
int i,j;
printf("after transpose");
for(j=1;j<=b;j++)
{
```

```
for(i=1;i<=a;i++)  
{  
    printf("\t%d",m[i][j]);  
}  
printf("\n");  
}  
}
```

INPUT:

enter the rows and columns of matrix..3 3

enter the elements

enter m[1][1]=1

enter m[1][2]=2

enter m[1][3]=3

enter m[2][1]=4

enter m[2][2]=5

enter m[2][3]=6

enter m[3][1]=7

enter m[3][2]=8

enter m[3][3]=9

OUTPUT:

before transpose 1 2 3

4 5 6

7 8 9

after transpose 1 4 7

2 5 8

3 6 9

d) Write C programs that use both recursive and non-recursive functions

i) To find the factorial of a given integer using non-recursive function.

AIM: To find factorial using non-recursion.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int factorial(int);
int a,r;
clrscr();
printf("Enter a number");
scanf("%d",&a);
r=factorial(a);
printf("The factorial value of %d is %d\n",a,r);
getch();
}
int factorial(int n)
{
int i,f=1;
for(i=1;i<=n;i++)
{
f=f*i;
}
return f;
}
```

INPUT:

Enter a number

5

OUTPUT:

The factorial value of 5 is 120

AIM: To find the factorial of a given integer using recursive function

```
#include <stdio.h>
#include <conio.h>
long int recursivefact(int n);
int main()
{
    int n;
    clrscr();
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Factorial of %d = %ld", n, recursivefact(n));
    return 0;
}
long int recursivefact(int n)
{
    if (n >= 1)
        return n*recursivefact(n-1);
    else
        return 1;
}
```

INPUT:

Enter a positive integer:6

OUTPUT:

Factorial of 6 = 720

ii. To find the GCD (greatest common divisor) of two given integers using non-recursive function.

AIM: To find GCD of two integers using non recursive function.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1,n2,GCD;
    clrscr();
    printf("Enter two positive integers: ");
    scanf("%d %d",&n1,&n2);
    GCD = gcd(n1,n2);
    printf("The GCD of two numbers %d and %d is %d.", n1, n2, GCD);
    getch();
}

int gcd(int a, int b)
{
    int i, gcd;
    for(i=1; i <= a && i <= b; ++i)
    {
        if(a%i==0 && b%i==0)
            gcd = i;
    }
    return gcd;
}
```

INPUT: Enter two positive integers 15 4

OUTPUT:

The GCD of two numbers 15 and 4 is 1

AIM: To find the GCD (greatest common divisor) of two given integers using recursive function

```
#include <stdio.h>
#include <conio.h>
int gcd(int n1, int n2);
int main()
{
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.", n1, n2, gcd(n1,n2));
    return 0;
}

int gcd(int n1, int n2)
{
    if (n2 != 0)
        return gcd(n2, n1%n2);
    else
        return n1;
}
```

Output:

Enter two positive integers: 12 16

G.C.D of 12 and 16 is 4.

iii. To find x^n using non-recursive function

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int base,pow,i,r=1,t;
    printf("Enter a number:");
    scanf("%d",&base);
    printf("Enter the power:");
    scanf("%d",&pow);
    for(i=1;i<=pow;i++)
    {
        t=base;
        r=r*t;
    }
    printf("Result:%d",r);
    getch();
}
```

Output:

Enter a number:2

Enter the power: 3

Result: 8

AIM: To find x^n using recursive function

```
#include<stdio.h>
#include<conio.h>
int power(int base, int powerRaised);
void main()
{
    int base, powerRaised, result;
    clrscr();
    printf("Enter base number: ");
    scanf("%d",&base);
    printf("Enter power number(positive integer): ");
    scanf("%d",&powerRaised);
    result = power(base, powerRaised);
    printf("%d^%d = %d", base, powerRaised, result);
    getch();
}

int power(int base, int powerRaised)
{
    if(powerRaised != 0)
        return (base*power(base, powerRaised-1));
    else
        return 1;
}
```

INPUT:

Enter base number: 3

Enter power number(positive integer): 4

OUTPUT:

3^4=81

e. Write a program for reading elements using pointer into array and display the values using array.

AIM: To read elements using pointer into array and display the values using array.

Source Code:

```
#include <stdio.h>
#include <conio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int N, i;
    int * ptr = arr;
    printf("Enter size of array: ");
    scanf("%d", &N);
    printf("Enter elements in array:\n");
    for (i = 0; i < N; i++)
    {
        scanf("%d", &ptr[i]);
    }
    printf("Array elements: ");
    for (i = 0; i < N; i++)
    {
        printf("%d, ", i[ptr]);
    }
    return 0;
}
```

Input:

Enter size of array: 10

Enter elements in array: 1 2 3 4 5 6 7 8 9 10

Output:

Array elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

f. Write a program for display values reverse order from array using pointer.

AIM: To display values reverse order from array using pointer.

Source Code:

```
#include<stdio.h>
#include<conio.h>
#define MAX 30
void main()
{
    int size, i, arr[MAX];
    int *ptr;
    clrscr();
    ptr = &arr[0];
    printf("\nEnter the size of array : ");
    scanf("%d", &size);
    printf("\nEnter %d integers into array: ", size);
    for (i = 0; i < size; i++)
    {
        scanf("%d", ptr);
        ptr++;
    }
    ptr = &arr[size - 1];
    printf("\nElements of array in reverse order are :");
    for (i = size - 1; i >= 0; i--) {
        printf("\nElement%d is %d : ", i, *ptr);
        ptr--;
    }
    getch();
}
```

Input:

Enter the size of array : 5

Enter 5 integers into array: 11 22 33 44 55

Output:

Elements of array in reverse order are :

Element 4 is: 55

Element 3 is: 44

Element 2 is: 33

Element 1 is: 22

Element 0 is: 11

g. Write a program through pointer variable to sum of n elements from array.

AIM: To find sum of n elements from array through pointer variable.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int numArray[10];
    int i, sum = 0;
    int *ptr;
    printf("\nEnter 10 elements : ");
    for (i = 0; i < 10; i++)
        scanf("%d", &numArray[i]);
    ptr = numArray;
    for (i = 0; i < 10; i++)
    {
        sum = sum + *ptr;
        ptr++;
    }
    printf("The sum of array elements : %d", sum);
}
```

Input:

Enter 10 elements: 11 12 13 14 15 16 17 18 19 20

Output:

The sum of array elements is 155

5. FILES

- a. Write a C program to display the contents of a file to standard output device.**

AIM: To display the contents of a file to standard output device.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fp;
    char ch;
    clrscr();
    fp=fopen("mrecw.txt","r");
    if(fp==NULL)
    {
        printf("file does not exist");
        exit(0);
    }
    else
        printf("file exist");
    while((ch=fgetc(fp))!=EOF)
    {
        printf("%c",ch);
    }
    fclose(fp);
    getch();
}
```

Input:

Mrecw.txt :welcome to mrecw

Output:

welcome to mrecw

- b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.

AIM: To copy one file to another, replacing all lowercase characters with their uppercase equivalents.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp1,*fp2;
char ch;
clrscr();
fp1=fopen("check.txt","r");
fp2=fopen("check1.txt","w");
if(fp1==NULL)
{
printf("files not exist");
exit(0);
}
else
printf("file exist");
while((ch=fgetc(fp1))!=EOF)
{
ch=toupper(ch);
fputc(ch,fp2);
}
fclose(fp1);
fclose(fp2);
printf("the resulted contents are\n");
fp2=fopen("check1.txt","r");
while((ch=fgetc(fp2))!=EOF)
{
printf("%c",ch);
}
fclose(fp2);
getch();
}
```


Input:

Check.txt : hello world

Output:

The resulted contents are

HELLO WORLD

- c. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.

AIM: To count the number of times a character occurs in a text file.

Source Code:

```
#include<stdio.h>
void main(int argc,char *argv[])
{
FILE *f1;
char ch;
int count=0;
if(argc!=3)
{
printf("error occurred");
exit();
}
f1=fopen(argv[1],"r");
while((ch=getc(f1))!=EOF)
{
if(ch==argv[2])
count++;
}
fclose(f1);
printf("character %c occurred %d times",ch,count);
}
```

Input:

a.txt: hello how are you

prg.exe a.txt O

Output:

character O occurred 3 times

- d. Write a C program that does the following: It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function) Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function) The program should then read all 10 values and print them back.

AIM: To store 10 elements in binary files

Source code:

```
#include<stdio.h>

/* Our structure */
struct rec
{
    int x,y,z;
};

int main(int argc,char *argv)
{
    int counter;
    FILE *ptr_myfile;
    struct rec my_record;

    ptr_myfile=fopen("test.bin","rb");
    if (!ptr_myfile)
    {
        printf("Unable to open file!");
        return 1;
    }

    fseek(ptr_myfile, sizeof(struct rec), SEEK_END);
    rewind(ptr_myfile);

    for ( counter=1; counter <= 10; counter++)
    {
        fread(&my_record,sizeof(struct rec),1,ptr_myfile);
        printf("%d\n",my_record.x);
    }
    fclose(ptr_myfile);
}
```

```
        return 0;  
    }
```

Input:

File.exe 10 20 34 45 67 44 33 78 90 56

Output:

My record 10 20 34 45 67 44 33 78 90 56

- e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

AIM: To merge two files into a third file.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp1,*fp2,*fp3;
char ch;
clrscr();
fp1=fopen("DATA1.txt","r");
fp2=fopen("DATA2.txt","r");
if(fp1==NULL||fp2==NULL)
{
printf("files not exist");
exit(0);
}
else
printf("file exist");
fp3=fopen("result.txt","w");
while((ch=fgetc(fp1))!=EOF)
{
fputc(ch,fp3);
}
while((ch=fgetc(fp2))!=EOF)
{
fputc(ch,fp3);
}
fclose(fp1);
fclose(fp2);
fclose(fp3);
printf("two files merged into third file");
printf("the resultant contents are\n");
fp3=fopen("result.txt","r");
while((ch=fgetc(fp3))!=EOF)
{
printf("%c",ch);
}
fclose(fp3);
getch();
}
```

Input:

Data1.txt: welcome

Data2.txt: to MRECW

Output:

two files merged into third file

the resultant contents are

result.txt: welcome to MRECW

6. STRINGS

- a. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.

AIM: To convert a Roman numeral ranging from I to L to its decimal equivalent.

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
    char rom[30];
    int a[30], l, i, k, dec;
    clrscr();
    printf("Enter the roman number\n");
    scanf("%s", &rom);
    l = strlen(rom);
    for(i = 0; i < l; i++)
    {
        switch (rom[i])
        {
            case 'I': a[i] = 1;
                    break;
            case 'V': a[i] = 5;
                    break;
            case 'X': a[i] = 10;
                    break;
            case 'L': a[i] = 50;
                    break;
            case 'C': a[i] = 100;
                    break;
            case 'D': dec = dec + 500;
                    break;
            case 'M': a[i] = 1000;
                    break;
            default : printf("Invalid choice");
                    break;
        }
    }
```

```
    }  
    k = a[l - 1];  
    for(i = l - 1; i > 0; i--)  
    {  
        if(a[i] > a[i - 1])  
        {  
            k = k - a[i - 1];  
        }  
        if(a[i] <= a[i - 1])  
        {  
            k = k + a[i - 1];  
        }  
    }  
    printf("decimal equivalent is %d", k);  
    getch();  
}
```

Input:

Enter the roman number: V

Output:

decimal equivalent is: 5

- b. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent.

AIM: To convert a number ranging from 1 to 50 to Roman equivalent.

Source Code:

```
#include<stdio.h>
void predigits(char c1,char c2);
void postdigits(char c,int n);

char roman_Number[1000];
int i=0;

int main(){

    int j;
    long int number;

    printf("Enter any natural number: ");
    scanf("%d",&number);

    if(number <= 0){
        printf("Invalid number");
        return 0;
    }

    while(number != 0){

        if(number >= 1000){
            postdigits('M',number/1000);
            number = number - (number/1000) * 1000;
        }
        else if(number >=500){
            if(number < (500 + 4 * 100)){
                postdigits('D',number/500);
                number = number - (number/500) * 500;
            }
            else{
                predigits('C','M');
                number = number - (1000-100);
            }
        }
        else if(number >=100){
            if(number < (100 + 3 * 100)){
                postdigits('C',number/100);
                number = number - (number/100) * 100;
            }
        }
    }
}
```

```
    }
    else{
        predigits('L','D');
        number = number - (500-100);
    }
}
else if(number >=50){
    if(number < (50 + 4 * 10)){
        postdigits('L',number/50);
        number = number - (number/50) * 50;
    }
    else{
        predigits('X','C');
        number = number - (100-10);
    }
}
else if(number >=10){
    if(number < (10 + 3 * 10)){
        postdigits('X',number/10);
        number = number - (number/10) * 10;
    }
    else{
        predigits('X','L');
        number = number - (50-10);
    }
}
else if(number >=5){
    if(number < (5 + 4 * 1)){
        postdigits('V',number/5);
        number = number - (number/5) * 5;
    }
    else{
        predigits('I','X');
        number = number - (10-1);
    }
}
else if(number >=1){
    if(number < 4){
        postdigits('I',number/1);
        number = number - (number/1) * 1;
    }
    else{
        predigits('I','V');
        number = number - (5-1);
    }
}
```

```
    }

    printf("Roman number will be: ");
    for(j=0;j<i;j++)
        printf("%c",roman_Number[j]);

    return 0;

}

void predigits(char c1,char c2){
    roman_Number[i++] = c1;
    roman_Number[i++] = c2;
}

void postdigits(char c,int n){
    int j;
    for(j=0;j<n;j++)
        roman_Number[i++] = c;

}
```

Input:

enter the natural number 23

Output:

Roman number will be XXIII

- c. Write a C program that uses functions to perform the following operations:

AIM: To insert a sub-string in to a given main string from a given position.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char a[30],b[30],c[30];
    int pos=0,i=0,l,la,lb,lc,j;
    clrscr();
    puts("enter a string");
    gets(a);
    puts("enter sub string");
    gets(b);
    puts("enter position for insertion");
    scanf("%d",&pos);
    la=strlen(a);
    lb=strlen(b);
    l=pos+lb;
    lc=la+lb;
    for(i=0;i<pos;i++)
    {
        c[i]=a[i];
    }
    j=0;
    for(i=pos;i<=l;i++)
    {
        c[i]=b[j];
        j++;
    }
    j=pos;
    for(i=l;i<lc;i++)
    {
        c[i]=a[j];
        j++;
    }
    c[i]='\0';
    puts("string after insertion is:");
    printf("%s",c);
    getch();
}
```

Input:

Enter the Original String: computer

Enter the sub String: gec

Enter the index where you want to insert the sub String: 3

Output:

The String after insertion: ----- comgecputer

AIM: To delete n Characters from a given position in a given string

Source Code:

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

void delchar(char *x,int a, int b);

void main()
{
    char string[10];
    int n,pos,p;
    clrscr();

    puts("Enter the string");
    gets(string);
    printf("Enter the position from where to delete");
    scanf("%d",&pos);
    printf("Enter the number of characters to be deleted");
    scanf("%d",&n);
    delchar(string, n,pos);
    getch();
}

/* Function to delete n characters*/
void delchar(char *x,int a, int b)
{
    if ((a+b-1) <= strlen(x))
    {
        strcpy(&x[b-1],&x[a+b-1]);
        puts(x);
    }
}
```

Input:

Enter the string: jayapal

Enter the position from where to delete:4

Enter the number of characters to be deleted 2

Output:

jayal

- d. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)

AIM: To determine if the given string is a palindrome or not.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
char str1[20],str2[20],str3[20],k;
clrscr();
printf("enter the string\n");
scanf("%s",str1);
strcpy(str3,str1);
strcpy(str2,strrev(str1));
k=strcmp(str3,str2);
if(k==0)
{
printf("the given string is palindrome");
}
else
{
printf("the given string is not a palindrome");
}
getch();
}
```

Input:

Enter the string: Madam

Output:

the given string is palindrome

- e. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn't contain ch.

AIM: To displays the position of a character ch in the string S or – 1 if S doesn't contain ch.

Source Code:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
    char s[30], t[20];
    char *found;
    clrscr();
    puts("Enter the first string: ");
    gets(s);
    puts("Enter the string to be searched: ");
    gets(t);
    found = strstr(s, t);
    if(found)
    {
        printf("Second String is found in the First String at %d position.\n",
            found - s);
    }
    else
    {
        printf("-1");
    }
    getch();
}
```

Input:

1. Enter the first string:
kali
Enter the string to be searched:
L

Output:

second string is found in the first string at 2 position

2. Enter the first string:
mrecw
Enter the string to be searched:
ma
-1

- f. Write a C program to count the lines, words and characters in a given text.

AIM: To count the lines, words and characters in a given text.

Source code:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char str[100];
    int i = 0, l = 0, f = 1;
    clrscr();
    puts("Enter any string\n");
    gets(str);
    for(i = 0; str[i] != '\0'; i++)
    {
        l = l + 1;
    }
    printf("The number of characters in the string are %d\n", l);
    for(i = 0; i <= l-1; i++)
    {
        if(str[i] == ' ')
        {
            f = f + 1;
        }
    }
    printf("The number of words in the string are %d", f);
    getch();
}
```

Input :

Enter any string

abc def ghi jkl mno pqr stu vwx yz

Output:

The number of characters in the string are 34

The number of words in the string are 9

7. MISCELLANEOUS

- a. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

AIM: To find the smallest, largest, sum and average using menu.

Source Code:

```
#include<stdio.h>
#include<conio.h>
int smallest(int a[10],int n);
int largest(int a[10],int n);
int sum(int a[10],int n);
int average(int a[10],int n);
void main()
{
    int a[10],n,i,ch,small,large,total,avg;
    clrscr();
    printf("enter the value of n");
    scanf("%d",&n);
    printf("enter n elements in an array");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
do
{
    printf("\t *****");
    printf("\n\tMENU\n");
    printf("\t*****");
    printf("\n\t(1)TO FIND SMALLEST");
    printf("\n\t(2)TO FIND LARGEST ");
    printf("\n\t(3)TO FIND SUM");
    printf("\n\t(4)TO FIND AVERAGE");
    printf("\n\t(0)EXIT");
    printf("\n\t*****");
    printf("\n\n\tEnter your choice:");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:
            small=smallest(a,n);
            printf("the smallest element in a list is %d",small);
            break;
```

```
        case 2:
            large= largest(a,n);
            printf("the largest element in list is %d",large);
            break;
        case 3:
            total=sum(a,n);
            printf("the sum of the elements in a list is %d",total);
            break;
        case 4:
            avg=average(a,n);
            printf("the average of element in a list is %d",avg);
            break;
        case 0:
            printf("\n Choice Terminated");
            exit();
            break;
        default:
            printf("\n Invalid Choice");
    }}while(1);
    getch();
}
int smallest( int a[10],int n)
{
    int min,i;
    min=a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]<min)
            min=a[i];
    }
    return min;
}
int largest( int a[10],int n)
{
    int max,i;
    max=a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]>max)
            max=a[i];
    }
    return max;
}
int sum( int a[10],int n)
{
    int s=0,i;
```

```
for(i=0;i<n;i++)
{
s=s+a[i];
}
return s;
}
int average( int a[10],int n)
{
int s=0,i;
for(i=0;i<n;i++)
{
s=s+a[i];
}
return (s/n);
}
```

INPUT:

Enter the value of n

5

Enter the elements :

20 16 3 56 7 1

MENU OPTIONS

```
***** 1.TO FIND SMALLEST
        2.TO FIND LARGEST
        3.TO FIND SUM
        4.TO FIND AVERAGE
        5.EXIT
```

ENTER UR CHOICE 1

The smallest element is 1

ENTER UR CHOICE 2

The largest element is 56

b. Write a C program to construct a pyramid of numbers as follows:

(i) 1
 1 2
 1 2 3
 1 2 3 4

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, rows;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=1; i<=rows; ++i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    getch();
}
```

Input:

Enter no of rows:4

Output:

1
1 2
1 2 3
1 2 3 4

(ii) *
 **

Source Code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, c, k;
    printf("Enter number of rows:\n");
    scanf("%d",&n);
    for ( c = 1 ; c <= n ; c++ )
    {
        for( k = 1 ; k <= c ; k++ )
            printf("*");
        printf("\n");
    }
    getch();
}
```

Input:

Enter number of rows:3

Output:

```
*
**
***
```

(iii) 1
 2 3
 4 5 6

Source Code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int rows, i, j, number= 1;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=1; i <= rows; i++)
    {
        for(j=1; j <= i; ++j)
        {
            printf("%d ", number);
            ++number;
        }
        printf("\n");
    }
    getch();
}
```

Input: Enter number of rows:3

Output:

1
2 3
4 5 6

(iv) 1
 2 2
 3 3 3
 4 4 4 4

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, j;
    int num;
    printf("Enter the number of Digits :");
    scanf("%d", &num);
    for (i = 0; i <= num; i++)
    {
        for (j = 0; j < i; j++)
        {
            printf("%d ", i);
        }
        printf("\n");
    }
    getch();
}
```

Output:

1
2 2
3 3 3
4 4 4 4

(v) *
 * *
 * * *
 * * * *
 * * * *
 * * *
 * *
 *

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, rows;
    clrscr();
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=1; i<=rows; ++i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("* ");
        }
        printf("\n");
    }
    for(i=rows; i>=1; --i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("* ");
        }
        printf("\n");
    }
    getch();
}
```

Input:

Enter no of rows:

Output:

```
*
* *
* * *
* * * *
* * * *
* * *
* *
*
```

8. SORTING AND SEARCHING

- a. Write a C program that uses non recursive function to search for a Key value in a given list of integers using linear search method.

AIM: To search for a key value in a given list of integers using linear search method

Source Code:

```
#include<stdio.h>
#include<conio.h>
int linear(int a[],int,int);
void main()
{
    int a[20],n,i,key,pos;
    clrscr();
    printf("enter the no. of elements u want in array\n");
    scanf("%d",&n);
    printf("enter %d elements\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("enter the key which u wnt to search\n");
    scanf("%d",&key);
    pos=linear(a,n,key);
    if(pos==-1)
        printf("element is not found");
    else
        printf("element is found at %d position",pos+1);
    getch();
}
int linear(int a[],int n,int key)
{
    int i,k=0;
    for(i=0;i<n;i++)
    {
        if(key==a[i])
        {
            k=i;
        }
    }
    if(k==0)
        return(-1);
    else
```

```
return(k);  
}
```

INPUT:

enter the no. of elements u want in array 5

enter 5 elements

56 87 4 34 21

enter the key which u want to search 4

OUTPUT:

element is found at 3 position

- b. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.

AIM: To search for a Key value in a given sorted list of integers using binary search method.

Source Code

```
#include<stdio.h>
#include<conio.h>
int binary(int a[],int,int);
void main()
{
    int a[20],i,n,key,pos;
    clrscr();
    printf("enter the no. of elements u want in the list\n");
    scanf("%d",&n);
    printf("enter the %d elements of array\n",n);
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("enter the key u want to search\n");
    scanf("%d",&key);
    pos=binary(a,n,key);
    if(pos==-1)
        printf("key is not found");
    else
        printf("key is found at %d position",pos+1);
    getch();
}

int binary(int a[],int n,int key)
{
    int low=0,high=n-1,mid;
    while(high>=low)
    {
        mid=(low+high)/2;
        if(key>a[mid])
            low=mid+1;
        else if(key<a[mid])
            high=mid-1;
        else
            return(mid);
    }
    return(-1);
}
```

INPUT:

enter the no. of elements u want in the list 5

enter 5 elements of array

14 45 57 85 96

enter the key which u want to search 85

OUTPUT:

key is found at 4 position

- c. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

AIM: To implement the Bubble sort method to sort a given list of integers in ascending order.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void bubble(int a[20],int);
void main()
{
    int a[20],n,i;
    clrscr();
    printf("enter the no. of elements to sort\n");
    scanf("%d",&n);
    printf("enter %d elements\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("before sorting,the elements are\n");
    for(i=0;i<n;i++)
    {
        printf("%3d",a[i]);
    }
    bubble(a,n);
    getch();
}

void bubble(int a[20],int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}
```

```
}  
}  
}  
printf("after sorting,the elements are\n");  
for(i=0;i<n;i++)  
{  
printf("%3d",a[i]);  
}  
}
```

Input:

enter the no. of elements to sort 5

enter 5elements 1 5 3 2 4

Output:

before sorting,the elements are 1 5 3 2 4

after sorting,the elements are 1 2 3 4 5

- d. Write a C program that sorts the given array of integers using selection sort in descending order.

AIM: To sort the given array of integers using selection sort in descending order.

Source Code:

```
#include <stdio.h>
void selection_sort();
int a[30], n;
void main()
{
    int i;
    printf("\nEnter size of an array: ");
    scanf("%d", &n);
    printf("\nEnter elements of an array:\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    selection_sort();
    printf("\n\nAfter sorting:\n");
    for(i=0; i<n; i++)
        printf("\n%d", a[i]);
    getch();
}
void selection_sort()
{
    int i, j, min, temp;
    for (i=0; i<n; i++)
    {
        min = i;
        for (j=i+1; j<n; j++)
        {
            if (a[j] > a[min])
                min = j;
        }
        temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
}
```

Input:

Enter size of an array: 6

Enter elements of an array: 3 1 4 2 5 6

Output:

After sorting:

6

5

4

3

2

1

- e. Write a C program that sorts the given array of integers using insertion sort in ascending order

AIM: To sort the given array of integers using insertion sort in ascending order.

Source Code:

```
#include<stdio.h>
#include<conio.h>
void insertion(int [], int );
int main()
{
    int arr[30];
    int i,size;
    printf("Enter total no. of elements : ");
    scanf("%d",&size);
    printf("\n Enter the elements to sort:");
    for(i=0; i<size; i++)
        scanf("%d",&arr[i]);
    insertion(arr,size);
    printf("\nAfter sorting\n");
    for(i=0; i<size; i++)
        printf(" %d",arr[i]);
    getch();
    return 0;
}
void insertion(int arr[], int size)
{
    int i,j,tmp;
    for(i=0; i<size; i++)
    {
        for(j=i-1; j>=0; j--)
        {
            if(arr[j]>arr[j+1])
            {
                tmp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=tmp;
            }
        }
        else
            break;
    }
}
```

```
}  
}  
}
```

Input:

Enter total no. of elements: 6

Enter the elements to sort:6 3 1 4 2 5

Output:

After sorting 1 2 3 4 5 6

- f. Write a C program that sorts a given array of names

AIM: To sort a given array of names.

Source Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
    int i, j, num;
    char name[20][10], t_name[15][10], temp[20];
    printf("Enter how many number of names to be sorted in alphabetical
order\n");
    scanf("%d", &num);
    printf("Please enter %d names one by one\n", num);
    for(i=0; i< num ; i++)
    {
        scanf("%s",name[i]);
        strcpy (t_name[i], name[i]);
    }
    for(i=0; i < num-1 ; i++)
    {
        for(j=i+1; j< num; j++)
        {
            if(strcmp(name[i],name[j]) > 0)
            {
                strcpy(temp,name[i]);
                strcpy(name[i],name[j]);
                strcpy(name[j],temp);
            }
        }
    }
    printf("Names before sorting in alphabetical order\n");
    for(i=0; i< num ; i++)
    {
        printf("%s\n",t_name[i]);
    }
    printf("Names after sorting in alphabetical order\n");
    for(i=0; i< num ; i++)
    {
```

```
        printf("%s\n",name[i]);  
    }  
}
```

Input:

Enter how many number of names to be sorted in alphabetical order 4

Please enter 4 names one by one

thiyagu

raja

mani

Arul

Output:

Entered names before sorting in alphabetical order

thiyagu

raja

mani

Arul

Entered names after sorting in alphabetical order

Arul

mani

raja

thiyagu

BEYOND THE SYLLABUS

1. Write a C program to find whether given number is perfect or not.

AIM: To find given number is perfect or not

Source Code:

```
#include<stdio.h>
int main(){
    int n,i=1,sum=0;
    printf("Enter a number: ");
    scanf("%d",&n);
    while(i<n){
        if(n%i==0)
            sum=sum+i;
        i++;
    }
    if(sum==n)
        printf("%d is a perfect number",i);
    else
        printf("%d is not a perfect number",i);
    return 0;}
```

Input:

Enter a number: 6

Output:

6 is a perfect number

2. Write a C program to find whether given number is Magic Number or not.

AIM: To find whether given number is Magic Number or not.

Source Code:

```
#include<conio.h>
#include<stdio.h>
void main()
{
    int t,i,s,res,digit,n,sum=0,rev=0;
    clrscr();
    printf("Enter number");
    scanf("%d",&n);
    t=n;
    while(n>0)
    {
        digit=n%10;
        sum=sum+digit;
        n=n/10;
    }
    s=sum;
    while(sum>0)
    {
        digit=sum%10;
        rev=rev*10+digit;
        sum=sum/10;
    }

    if(t==(s*rev))
        printf("%d is a magic number",t);
    else
        printf("%d is not a magic number");
    getch();
}
```

Input: Enter Number: 1729

Output: 1729 is a magic number

3. Write a C program to implement call by value and call by reference.**Call by Value:**

```
#include<stdio.h>
void interchange(int number1,int number2)
{
    int temp;
    temp = number1;
    number1 = number2;
    number2 = temp;
}
int main() {
int num1=50,num2=70;
    interchange(num1,num2);
    printf("\nNumber 1 : %d",num1);
    printf("\nNumber 2 : %d",num2);
    return(0);
}
```

OUTPUT:**Number 1 : 70****Number 2 : 50****Call by Reference:**

```
#include<stdio.h>
void interchange(int *num1,int *num2)
{
    int temp;
    temp = *num1;
    *num1 = *num2;
    *num2 = temp;
}
```

```
int main()
{
    int num1=50,num2=70;
    interchange(&num1,&num2);
    printf("\nNumber 1 : %d",num1);
    printf("\nNumber 2 : %d",num2);

    return(0);
}
```

OUTPUT:**Number 1 : 70****Number 2 : 50**

4. Write a C program to find sum of n elements given by user using malloc() and calloc() functions.

Using malloc()

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

int main()
{
    int num, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &num);
    ptr = (int*) malloc(num * sizeof(int)); //memory allocated using malloc
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }

    printf("Enter elements of array: ");

    for(i = 0; i < num; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```

OUTPUT:

```
Enter number of elements:5
Enter elements of array:
25    14    7    4    1
Sum=51
```

Using calloc()

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

int main()
{
    int num, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &num);

    ptr = (int*) calloc(num, sizeof(int));
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }

    printf("Enter elements of array: ");
    for(i = 0; i < num; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);

    free(ptr);
    return 0;
}
```

OUTPUT:

```
Enter number of elements:5
Enter elements of array:
25    14    7    4    1
Sum=51
```

5. Write a C program using structures to display the details of five Students.

Source Code:

```
#include <stdio.h>
struct student
{
    char name[50];
    int roll;
    float marks;
} s[10];

int main()
{
    int i;

    printf("Enter information of students:\n");

    // storing information
    for(i=0; i<10; ++i)
    {
        s[i].roll = i+1;

        printf("\nFor roll number%d,\n",s[i].roll);

        printf("Enter name: ");
        scanf("%s",s[i].name);

        printf("Enter marks: ");
        scanf("%f",&s[i].marks);

        printf("\n");
    }
    printf("Displaying Information:\n\n");
    // displaying information
    for(i=0; i<10; ++i)
    {
        printf("\nRoll number: %d\n",i+1);
        printf("Name: ");
        puts(s[i].name);
        printf("Marks: %.1f",s[i].marks);
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

Enter information of students:

For roll number1,

Enter name: Tom

Enter marks: 98

For roll number2,

Enter name: Jerry

Enter marks: 89

.

.

.

Displaying Information:

Roll number: 1

Name: Tom

Marks: 98

SAMPLE PROJECT CODE**/* STUDENT INFORMATION SYSTEM*/**

```
#include<conio.h>
#include<stdio.h>
void menu();
void header();
void newfile();
void add();
void search();
void display();
void deleted();
void update();
int ch,f=0;
char tid[10];
struct student
{
char id[10],name[10],rollno[10],branch[10],sec[10],addr[10],dob[10];
}rec;
struct student rec1={0};
void main()
{
menu();
}
void menu()
{
header();
gotoxy(23,9);
cprintf("\n1.CREATE A NEW FILE");
gotoxy(23,10);
cprintf("\n2.ADD A NEW RECORD");
gotoxy(23,11);
cprintf("\n3.SEARCH A RECORD");
gotoxy(23,12);
cprintf("\n4.DISPLAY A RECORD");
gotoxy(23,13);
cprintf("\n5.DELETE A RECORD");
gotoxy(23,14);
cprintf("\n6.UPDATE A RECORD");
gotoxy(23,15);
cprintf("\n7.EXIT");
gotoxy(23,16);
cprintf("\nENTER UR CHOICE");
scanf("%d",&ch);
switch(ch)
```

```
{
case 1:newfile();
    break;
case 2:add();
    break;
case 3:search();
    break;
case 4:display();
    break;
case 5:deleted();
    break;
case 6:update();
    break;
case 7:exit(0);
}
getch();
}
void newfile()
{
FILE *fp;
fp=fopen("student.txt","w");
printf("\n ONE NEW FILE CREATED SUCCESFULLY");
getch();
fclose(fp);
getch();
menu();
}

void add()
{
FILE *fp;
fp=fopen("student.txt","a+");
if(fp==NULL)
{
printf("FILE IS NOT OPENED");
getch();
exit(0);
}
printf("\n ENTER STUDENT ID");
scanf("%s",&rec.id);
printf("\n ENTER STUDENT NAME");
scanf("%s",&rec.name);
printf("\n ENTER STUDENT ROLL NO");
scanf("%s",&rec.rollno);
printf("\n ENTER BRANCH NAME");
scanf("%s",&rec.branch);
```

```
printf("\n ENTER SECTION");
scanf("%s",&rec.sec);
printf("\n ENTER ADDRESS");
scanf("%s",&rec.addr);
fwrite(&rec,sizeof(rec),1,fp);
printf("\nONE NEW RECORD CREATED SUCCESFULLY");
getch();
fclose(fp);
menu();
}

void search()
{
FILE *fp;
fp=fopen("student.txt","r+");
if(fp==NULL)
{
printf("FILE IS NOT OPENED");
getch();
exit(0);
}
printf("ENTER UR ID TO BE SEARCHD:");
scanf("%s",&tid);
while(fread(&rec,sizeof(rec),1,fp))
{
if(!strcmp(rec.id,tid))
{
f=1;
printf("\n\t\t\tRECORD IS FOUND");
printf("\nID :%s",rec.id);
printf("\nNAME :%s",rec.name);
printf("\nROLL NO:%s",rec.rollno);
printf("\nBRANCH :%s",rec.branch);
printf("\nSECTION:%s",rec.sec);
printf("\nADDRESS:%s",rec.addr);
printf("\nDOB:%s",rec.dob);
printf("\n\n");
getch();
}
}
if(f==0)
printf("\n\t\t\tRECORD is not FOUND");
getch();
fclose(fp);
menu();
}
```



```
void display()
{
    FILE *fp;
    fp=fopen("student.txt","r+");
    if(fp==NULL)
    {
        printf("FILE IS NOT OPENED");
        getch();
        exit(0);
    }
    while(fread(&rec,sizeof(rec),1,fp))
    {
        if(strcmp(rec.id,'\0'))
        {
            f=1;
            printf("\nID :%s",rec.id);
            printf("\nNAME :%s",rec.name);
            printf("\nROLL NO :%s",rec.rollno);
            printf("\nBRANCH :%s",rec.branch);
            printf("\nSECTION :%s",rec.sec);
            printf("\nADDRESS :%s",rec.addr);
            printf("\nDOB :%s",rec.dob);
            printf("\n\n");
            getch();
        }
    }
    if(f==0)
        printf("\n\n\t\t\tNO RECORDS");
    getch();
    fclose(fp);
    menu();
}

void deleted()
{
    FILE *fp;
    fp=fopen("student.txt","r+");
    if(fp==NULL)
    {
        printf("FILE IS NOT OPENED");
        getch();
        exit(0);
    }
    printf("ENTER UR ID TO BE DELETED:");
    scanf("%s",&tid);
```

```
while(fread(&rec,sizeof(rec),1,fp))
{
    if(!strcmp(rec.id,tid))
    {
        f=1;
        printf("\n\t\tONE RECORD IS DELETED SUCCESSFULLY");
        printf("\nID :%s",rec.id);
        printf("\nNAME :%s",rec.name);
        printf("\nROLL NO:%s",rec.rollno);
        printf("\nBRANCH :%s",rec.branch);
        printf("\nSECTION:%s",rec.sec);
        printf("\nADDRESS:%s",rec.addr);
        printf("\nDOB:%s",rec.dob);
        printf("\n\n");
        getch();
        fseek(fp,ftell(fp)-sizeof(rec),0);
        fwrite(&rec1,sizeof(rec),1,fp);
    }
}
if(f==0)
printf("\n\n\t\tRECORD is not FOUND");
getch();
fclose(fp);
menu();
}

void update()
{
    FILE *fp;
    fp=fopen("student.txt","r+");
    if(fp==NULL)
    {
        printf("FILE IS NOT OPENED");
        getch();
        exit(0);
    }
    printf("ENTER UR ID TO BE UPDATED:");
    scanf("%s",&tid);
    while(fread(&rec,sizeof(rec),1,fp))
    {
        if(!strcmp(rec.id,tid))
        {
            f=1;
            printf("\nID :%s",rec.id);
            printf("\nNAME :%s",rec.name);
            printf("\nROLL NO:%s",rec.rollno);
```

```
printf("\nBRANCH :%s",rec.branch);
printf("\nSECTION:%s",rec.sec);
printf("\nADDRESS:%s",rec.addr);
printf("\nDOB:%s",rec.dob);
printf("\n\n");
getch();printf("1.NAME\n2.ROLL NO \n3.BRANCH NAME
\n4.SECTIO \n5.ADDRESS\n6.DOB:");
printf("enter ur choice :");
scanf("%d",&ch);

switch(ch)
{
case 1:printf("enter new name");
scanf("%s",rec.name);
break;
case 2:printf("enter new roll no");
scanf("%d",&rec.rollno);
break;
case 3:printf("enter new branch");
scanf("%s",rec.branch);
break;
case 4:printf("enter new section");
scanf("%s",rec.sec);
break;
case 5:printf("enter new address");
scanf("%s",rec.addr);
break;
}
fseek(fp,ftell(fp)-sizeof(rec),0);
fwrite(&rec,sizeof(rec),1,fp);
printf("\n\t\tONE RECORD IS updated SUCCESSFULLY");
}
}
if(f==0)
printf("\n\t\tRECORD is not FOUND");
getch();
fclose(fp);
menu();
}

void header()
{
clrscr();
gotoxy(20,5);
textcolor(BLUE);
cprintf("*****");
```

```
gotoxy(10,6);
textcolor(YELLOW);
cprintf("@@@@@@@@@@@@@@....STUDENT
MANAGEMENT....@@@@@@@@@@@@@@@@");
gotoxy(20,7);
textcolor(BLUE);
cprintf("*****");
}
```