```
//write a c++ program for addition of matrix
#include<iostream.h>
#include<conio.h>
void main()
{
int a[2][2],b[2][2],c[2][2];
int i,j;
 clrscr();
cout<<"Enter first matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>a[i][j];
}
}
cout<<"Enter second matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>b[i][j];
}
}
cout<<"Element of first matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<a[i][j]<<"\n";
}
}
cout<<"Element of second matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<b[i][j];
}
}
cout<<"\naddition of matrix is:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 c[i][j]=a[i][j]+b[i][j];
}
}
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<c[i][j];
}
}
getch();
}
```

Output:-

```
Enter first matrix:-1 2 3 4
Enter second matrix:-1 2 3 4
Element of first matrix:- 1 2 3 4
Element of second matrix:- 1 2 3 4
addition of matrix is:-2 4 6 8
```

```
//write a c++ program for Subtraction of matrix
#include<iostream.h>
#include<conio.h>
void main()
{
int a[2][2],b[2][2],c[2][2];
int i,j;
 clrscr();
cout<<"Enter first matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>a[i][j];
}
}
cout<<"Enter second matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>b[i][j];
}
}
cout<<"Element of first matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<a[i][j]<<"\n";
}
}
cout<<"Element of second matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<b[i][j];
}
}
cout<<"\nSubtraction of matrix is:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 c[i][j]=a[i][j]-b[i][j];
}
}
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<c[i][j];
}
}
getch();
}
```

Output:-

```
Enter first matrix:-4 3 5 6
Enter second matrix:-1 2 3 4
Element of first matrix:- 4 3 5 6
Element of second matrix:- 1 2 3 4
Subtraction of matrix is:-3 1 2 2
```

```
// write a c++ program for Multiplication of matrix
#include<iostream.h>
#include<conio.h>
void main()
{
int a[2][2],b[2][2],c[2][2],k;
int i,j;
 clrscr();
cout<<"Enter first matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>a[i][j];
}
}
cout<<"Enter second matrix:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cin>>b[i][j];
}
}
cout<<"Element of first matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<a[i][j]<<"\n";
}
}
cout<<"Element of second matrix:-\n";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<b[i][j];
}
}
cout<<"\nMultiplication of matrix is:-";
for(i=0;i<2;i++){
for(j=0;j<2;j++){
c[i][j]=0;
 for(k=0;k<2;k++){
 c[i][j]+=a[i][j]*b[i][j];
}
}
}
for(i=0;i<2;i++){
for(j=0;j<2;j++){
 cout<<c[i][j];
}
}
getch();
}
```

```
//write a program to demonstrate traversal operation.

#include<iostream.h>

#include<conio.h>

void printArray(int* arr, int n)

{

int i;

cout << "Array: ";

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

{

cout << arr[i] << " ";

}

}

// Driver code

int main()

{

int arr[] = {2, -1, 5, 6, 0, -3};

int n = sizeof(arr) / sizeof(arr[0]);

printArray(arr, n);

return 0;

}
```

```
//write a program to insert element at last position.
#include<iostream.h>
#include<conio.h>
void main()
{
int arr=[20],i,n;
clrscr();
cout<<"Enter array element:-";
for(i=0;i<5;i++)
{
cout<<arr[i];
}
cout<<"\nEnter number to insert:-";
cin>>n;
for(i=0;i<5;i++)
{
cout<<arr[i]<<" ";
}
cout<<endl;
getch();
}
```

output:-

```
Enter array element:-1 2 3 1 4
Enter number to insert:-5
New array is:-1 2 3 1 4 5
```

```
//write a program to insert an element in array at given position.
#include<iostream.h>
#include<conio.h>
void main()
{
int arr[20],i,n,pos,size;
clrscr();
cout<<"enter the size of array:-";
cin>>size;
cout<<"enter "<<size<<" arry element:-";
for(i=0;i<size;i++)
{
cin>>arr[i];
}
cout<<"\nenter the element to insert:-";
cin>>n;
cout<<"at what position:-";
cin>>pos;
for(i=size;i>pos;i--)
{
arr[i]=arr[i-1];
}
arr[i]=n;
size++;
cout<<"\nthe new array:-\n";
for(i=0;i<size;i++)
{
```

```
cout<<arr[i]<<" ";

}

cout<<endl;

getch();

}
```

output:-

```
enter the size of array:-1

enter 1 arry element:-2

enter the element to insert:-1

at what position:-1

the new array:-2 1
```

```
//write a program for deletion of array element

#include<iostream.h>

#include<conio.h>

void main()

{

int arr[20],size,i,j,elem;

clrscr();

cout<<"enter size of array:-";

cin>>size;

cout<<"enter element of an array:-"<<endl;

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

{

cin>>arr[i];

}

cout<<"enter element you want to delete:-";

cin>>elem;

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

{

if(arr[i]==elem)

{

for(j=i;j<size-1;j++)

{

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

}

}

}

cout<<"element in array:-";
for(i=0;i<size-1;i++)

{

cout<<arr[i]<<" ";

}

getch();

}
```

output:-

```
enter size of array:-5

enter element of an array:- 1 2 3 4 5

enter element you want to delete:-2

element in array:-1 3 4 5
```

```
//write a program to demonstrate a stack.

#include<iostream.h>

#include<conio.h>

int stack[100],n=100,top=-1;

void push(int val){

if(top>=n-1){

cout<<"Stack overflow"<<endl;

}

else{

top++;

stack[top]=val;

}

}

void pop(){

if(top<=-1){

cout<<"Stack underflow"<<endl;

}

else{

cout<<"The popped element is:-"<<stack[top]<<endl;

}

}

void display(){

if(top>=0){

cout<<"stack elements are:-";

for(int i=top;i>=0;i--){

cout<<stack[i]<<" ";

}
```

```
cout<<endl;

}

else{

cout<<"stack is empty";

}

}

int main(){

int ch,val;

clrscr();

cout<<"1.push in stack:-"<<endl;

cout<<"2.pop in stack:-"<<endl;

cout<<"3.display in stack:-"<<endl;

cout<<"4.exit"<<endl;

do{

cout<<"Enter choice:- "<<endl;

cin>>ch;

switch(ch){

case 1:{

cout<<"Enter valued to be push:-"<<endl;

cin>>val;

push(val);

break;

}

case 2:{

pop();

break;}

case 3:{
```

```
display();

break;

}

case 4:{

cout<<"exit"<<endl;

break;

}

default:{

cout<<"invalid choice"<<endl;

}

}

}

while(ch!=4);

return 0;

}
```

```
//Write a program to demonstrate the simple queue

#include <iostream.h>

#include<conio.h>

int queue[50];

int n = 50;

int front = - 1;

int rear = - 1;

void insert_ele() {

 int val;

 if (rear == n - 1)

 cout<<"Queue Overflow"<<endl;

 else {

 front = 0;

 cout<<" insert value in the queue : "<<endl;

 cin>>val;

 rear++;

 queue[rear] = val;

 }

}

void delete_ele() {

 if (front == - 1) {

 cout<<"Queue Underflow ";

 return ;

 } else {

 cout<<"Element deleted from queue is : "<< queue[front] <<endl;

 front++;;

 }
}
void display_queue () {
 if (front == - 1 )
 cout<<"Queue is empty"<<endl;
 else {
 cout<<"Queue elements are : ";
 for (int i = front; i <= rear; i++)
 cout<<queue[i]<<" ";
 cout<<endl;
 }
}
int main()
{
clrscr();
int ch;
 cout<<"1) insertion element to the queue"<<endl;
 cout<<"2) Delete element from queue"<<endl;
 cout<<"3) Display all the elements of queue"<<endl;
 cout<<"4) Exit"<<endl;
do {
 cout<<"Enter your choice : "<<endl;
 cin>>ch;
 switch (ch) {
 case 1: insert_ele();
 break;
 case 2: delete_ele();
 break;
 case 3: display_queue ();

 break;

 case 4: cout<<"Exit"<<endl;

 break;

 default: cout<<"Invalid choice"<<endl;

 }

 }

while(ch!=4);

 return 0;

getch();

}
```

```
// single linked list.
#include <iostream.h>
#include<conio.h>

// Node class to represent
// a node of the linked list.
class Node {
public:
   int data;
   Node* next;

   // Default constructor
   Node()
   {
            data = 0;
            next = NULL;
   }

   // Parameterised Constructor
   Node(int data)
   {
            this->data = data;
            this->next = NULL;
   }
};

// Linked list class to
// implement a linked list.
class Linkedlist {
   Node* head;

public:
   // Default constructor
   Linkedlist() { head = NULL; }

   // Function to insert a
   // node at the end of the
   // linked list.
   void insertNode(int);

   // Function to print the
   // linked list.
   void printList();

   // Function to delete the
   // node at given position
   void deleteNode(int);
};

// Function to delete the
// node at given position
void Linkedlist::deleteNode(int nodeOffset)
{
   Node *temp1 = head, *temp2 = NULL;
```

```
int ListLen = 0;

if (head == NULL) {
    cout << "List empty." << endl;
    return;
}

// Find length of the linked-list.
while (temp1 != NULL) {
    temp1 = temp1->next;
    ListLen++;
}

// Check if the position to be
// deleted is greater than the length
// of the linked list.
if (ListLen < nodeOffset) {
    cout << "Index out of range"
      << endl;
    return;
}

// Declare temp1
temp1 = head;

// Deleting the head.
if (nodeOffset == 1) {
```

```
            // Update head
            head = head->next;
            delete temp1;
            return;
    }

    // Traverse the list to
    // find the node to be deleted.
    while (nodeOffset-- > 1) {

            // Update temp2
            temp2 = temp1;

            // Update temp1
            temp1 = temp1->next;
    }

    // Change the next pointer
    // of the previous node.
    temp2->next = temp1->next;

    // Delete the node
    delete temp1;
}

// Function to insert a new node.
```

```
void Linkedlist::insertNode(int data)

{

   // Create the new Node.

   Node* newNode = new Node(data);



   // Assign to head

   if (head == NULL) {

            head = newNode;

            return;

   }



   // Traverse till end of list

   Node* temp = head;

   while (temp->next != NULL) {



            // Update temp

            temp = temp->next;

   }



   // Insert at the last.

   temp->next = newNode;

}



// Function to print the

// nodes of the linked list.

void Linkedlist::printList()

{
    Node* temp = head;

    // Check for empty list.
    if (head == NULL) {
            cout << "List empty" << endl;
            return;
    }

    // Traverse the list.
    while (temp != NULL) {
            cout << temp->data << " ";
            temp = temp->next;
    }
}

// Driver Code
void main()
{
    Linkedlist list;

    // Inserting nodes
    list.insertNode(1);
    list.insertNode(2);
    list.insertNode(3);
    list.insertNode(4);

    cout << "Elements of the list are: ";
    // Print the list

    list.printList();

    cout << endl;


    // Delete node at position 2.

    list.deleteNode(2);


    cout << "Elements of the list are: ";

    list.printList();

    cout << endl;

    getch();

}
```

```
//Write a cpp program for bubble sort.

#include<iostream.h>

#include<conio.h>

int main()

{

int arr[50],i,j,size,temp;

clrscr();

cout<<"enter the size of array";

cin>>size;

cout<<"enter the element os array";

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

{

cin>>arr[i];

}

cout<<"your array is";

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

{

cout<<arr[i]<<endl;

}

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

{

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

{

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

{

temp=arr[j];

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

}

}

}

cout<<"after bubble sorting ypou array is";

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

{

cout<<arr[i]<<endl;

}

getch();

}
```

output:-

```
enter the size of array5

enter the element os array 3 31 23 94 4

your array is 3 31 23 94 4

after bubble sorting you array is3 4 23 31 94
```

```
//write a program to find factorial of given number using recursion
#include<iostream.h>
#include<conio.h>
int fact(int num)
{
if(num==1){
return num;
}
return num*fact(num-1);
}
void main()
{
clrscr();
int n,result;
cout<<"Enter number:-";
cin>>n;
result=fact(n);
cout<<"The factorial of "<<n<<" is "<<result;
getch();
}
```

```
Enter number:-3
The factorial of 3 is 6
```

```
//write a program to print fibonacci series using recursion
#include<iostream.h>
#include<conio.h>
int fibo(int n)
{
if(n<=1){
return n;
}
return fibo(n-1)+fibo(n-2);
}
void main()
{
clrscr();
int num;
cout<<"ënter number:-";
cin>>num;
for(int i=0;i<num;i++){
cout<<fibo(i)<<" ";
}
getch();
}
```

output:

```
enter number:-4
0 1 1 2
```

```
//Reverse Array Using Recursion
#include<iostream.h>
#include<conio.h>
int reverse(int arr[], int start, int end)
{
 int temp;
 if(start < end) {
temp = arr[start];
arr[start] = arr[end];
arr[end] = temp;
// recursive function call
reverse(arr, start+1, end-1);
 }
 return 0;
}
void main() {
clrscr();
int n, arr[100], i;
cout << "Enter the size of an array \n";
cin >> n;
cout << "Enter an element of an array \n";
for(i = 0; i < n; i++)
{
cin >> arr[i];
}
reverse(arr, 0, n-1);
cout << "Reverse of an array is \n";
for(i = 0; i < n; i++)

{

cout << arr[i] << " ";

}

getch();

}
```

```
//Write a cpp program to demonstrate the linear search
#include<iostream.h>
#include<conio.h>
void main()
{
int num[]={12,6,34,3,45,4,1};
int x,i,f;
clrscr();
cout<<"Enter array:-";
for(i=0;i<7;i++){
cout<<num[i]<<" ";
}
cout<<"\nEnter number to search:-";
cin>>x;
f=0;
for(i=0;i<7;i++){
if(x==num[i]){
cout<<"Number found at index:-"<<i;
f=1;
break;
}
}
if(f==0){
cout<<"Number not found";
}
getch();
}
```

output:-

```
Enter array:-12 6 34 3 45 4 1

Enter number to search:-1

Number found at index:-6
```

```
//write a cpp program to demonstrate the binary search

#include<iostream.h>

#include<conio.h>

void main()

{

int num[]={2,5,9,13,22,45,89};

int x,f,s,m,e,i;

clrscr();

cout<<"Array:-";

for(i=0;i<7;i++){

cout<<num[i]<<" ";

}

cout<<"Enter the num you want to search:-";

cin>>x;

f=0;

s=0;

e=6;

while(s<=e){

m=(s+e)/2;

if(x==num[m]){

cout<<"Number found at index:-"<<m;

f=1;

break;

}

else if(x>num[m]){

s=m+1;
}
else if(x<num[m]){
e=m-1;
}
}
if(f==0){
cout<<"number not found";
}
getch();}
```

output:-

```
Array:-2 5 9 13 22 45 89
Enter the num you want to search:-9
Number found at index:-2
```

```
//write a program for Tower of Hanoi using recursion.
// C++ recursive function to
// solve tower of hanoi puzzle
#include <bits/stdc++.h>
using namespace std;

void towerOfHanoi(int n, char from_rod, char to_rod,
 char aux_rod)
{
 if (n == 0) {
 return;
 }
 towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
 cout << "Move disk " << n << " from rod " << from_rod
 << " to rod " << to_rod << endl;
 towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
}

// Driver code
int main()
{
 int N = 3;

 // A, B and C are names of rods
 towerOfHanoi(N, 'A', 'C', 'B');
 return 0;
}
```

C++ Program to Implement Adjacency Matrix:

```
#include<iostream.h>

#include<conio.h>

int vertArr[20][20]; //the adjacency matrix initially 0

int count = 0;

void displayMatrix(int v) {

  int i, j;

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

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

          cout << vertArr[i][j] << " ";

    }

    cout << endl;

  }

}

void add_edge(int u, int v) {      //function to add edge into the matrix

  vertArr[u][v] = 1;

  vertArr[v][u] = 1;

}

void main() {

  int v = 6;    //there are 6 vertices in the graph

  add_edge(0, 4);

  add_edge(0, 3);

  add_edge(1, 2);

  add_edge(1, 4);

  add_edge(1, 5);

  add_edge(2, 3);

  add_edge(2, 5);
  add_edge(5, 3);

  add_edge(5, 4);

  displayMatrix(v);

  getch();

}
```