1. **Write a program to store the elements in 1-D array and perform the operations like searching, sorting and reversing the elements.**

 **C++ Program - Reverse Array**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

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

 cout<<"Enter array size : ";

 cin>>size;

 cout<<"Enter array elements : ";

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

 {

 cin>>arr[i];

 }

 j=i-1; // now j will point to the last element

 i=0; // and i will be point to the first element

 while(i<j)

 {

 temp=arr[i];

 arr[i]=arr[j];

 arr[j]=temp;

 i++;

 j--;

 }

 cout<<"Now the Reverse of the Array is : \n";

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

 {

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

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following output:



1. **C++ Program - Linear Search in Array**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

 int arr[10], i, num, n, c=0, pos;

 cout<<"Enter the array size : ";

 cin>>n;

 cout<<"Enter Array Elements : ";

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

 {

 cin>>arr[i];

 }

 cout<<"Enter the number to be search : ";

 cin>>num;

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

 {

 if(arr[i]==num)

 {

 c=1;

 pos=i+1;

 break;

 }

 }

 if(c==0)

 {

 cout<<"Number not found..!!";

 }

 else

 {

 cout<<num<<" found at position "<<pos;

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **C++ Program - Sort Elements of Array in Ascending Order**

#include<iostream.h>

#include<conio.h>

**void** main()

{

**int** i,a[10],temp,j;

clrscr();

cout<<"Enter any 10 num in array: \n";

**for**(i=0;i<=10;i++)

{

cin>>a[i];

}

cout<<"\nData before sorting: ";

**for**(j=0;j<10;j++)

{

cout<<a[j];

}

**for**(i=0;i<=10;i++)

{

**for**(j=0;j<=10-i;j++)

{

**if**(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

cout<<"\nData after sorting: ";

**for**(j=0;j<10;j++)

{

cout<<a[j];

}

getch();

}

**Output**

Enter any 10 num in array:

2 5 1 7 5 3 8 9 11 4

Data After Sorting: 1 2 3 4 5 7 8 9 11



### C++ Program - Sort an Array Elements in Descending Order.

For Sort Elements of Array in Descending Order we print all Elements of array from last index to first. For example arr[10], arr[9], arr[8],....arr[0]

Example

#include<iostream.h>

#include<conio.h>

**void** main()

{

**int** i,a[10],temp,j;

clrscr();

cout<<"Enter any 10 num in array : \n";

**for**(i=0;i<=10;i++)

{

cin>>a[i];

}

cout<<"\n\nData before sorting: ";

**for**(j=0;j<10;j++)

{

cout<<a[j];

}

**for**(i=0;i<=10;i++)

{

**for**(j=0;j<=10-i;j++)

{

**if**(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

cout<<"\nData after sorting: ";

**for**(j=9;j>=0;j--)

{

cout<<a[j];

}

getch();

}

**Output**

Enter any 10 num in array:

2 5 1 7 5 3 8 9 11 4

Data After Sorting: 11 9 8 7 5 4 3 2 1



1. **Read the two arrays from the user and merge them and display the elements in sorted order.**

#include<iostream.h>

#include<conio.h>

void main()

 {

 int a[10],b[10],c[20],i;

 clrscr();

 cout<<"Enter Elements in 1st Array: ";

 for(i=0;i<10;i++)

 {

 cin>>a[i];

 }

 cout<<"Enter Elements in 2nd Array: ";

 for(i=0;i<10;i++)

 {

 cin>>b[i];

 }

 cout<<"\nElements of Array After Merge: ";

 for(i=0;i<10;i++)

 {

 c[i]=a[i];

 c[i+10]=b[i];

 }

 for(i=0;i<20;i++)

 {

 cout<<c[i];

 }

 getch();

 }}

**Output:**

1. **Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven]**

**C++ Program - Add Two Matrices**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

 int mat1[3][3], mat2[3][3], i, j, mat3[3][3];

 cout<<"Enter matrix 1 elements :";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cin>>mat1[i][j];

 }

 }

 cout<<"Enter matrix 2 elements :";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cin>>mat2[i][j];

 }

 }

 cout<<"Adding the two matrix to form the third matrix .....\n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 mat3[i][j]=mat1[i][j]+mat2[i][j];

 }

 }

 cout<<"The two matrix added successfully...!!";

 cout<<"The new matrix will be :\n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cout<<mat3[i][j]<<" ";

 }

 cout<<"\n";

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **C++ Program - Multiply Two Matrices**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

 int mat1[3][3], mat2[3][3], mat3[3][3], sum=0, i, j, k;

 cout<<"Enter first matrix element (3\*3) : ";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cin>>mat1[i][j];

 }

 }

 cout<<"Enter second matrix element (3\*3) : ";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cin>>mat2[i][j];

 }

 }

 cout<<"Multiplying two matrices...\n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 sum=0;

 for(k=0; k<3; k++)

 {

 sum = sum + mat1[i][k] \* mat2[k][j];

 }

 mat3[i][j] = sum;

 }

 }

 cout<<"\nMultiplication of two Matrices : \n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cout<<mat3[i][j]<<" ";

 }

 cout<<"\n";

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **C++ Program - Transpose Matrix**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

 int arr[3][3], i, j, arrt[3][3];

 cout<<"Enter 3\*3 Array Elements : ";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cin>>arr[i][j];

 }

 }

 cout<<"Transposing Array...\n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 arrt[i][j]=arr[j][i];

 }

 }

 cout<<"Transpose of the Matrix is :\n";

 for(i=0; i<3; i++)

 {

 for(j=0; j<3; j++)

 {

 cout<<arrt[i][j];

 }

 cout<<"\n";

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **Write a program to create a single linked list and display the node elements.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

class slink

{

private:

struct node

{

int data;

node \*next;

}

\*head;

public:

slink()

{

head=NULL;

}

void create();

void display();

};

void slink::create()

{

head=new node;

cout<<"Enter the data for the node:";

cin>>head->data;

head->next=NULL;

}

void slink::display()

{

node \*list;

list=head;

if(list==NULL)

cout<<"List is empty";

else

{

cout<<"The list is: ";

while(list!=NULL)

{

cout<<list->data<<"<==>";

list=list->next;

}

cout<<"NULL";

}

getch();

}

void main()

{

int option=0;

slink s;

while(option!=3)

{

clrscr();

cout<<"1.Create()"<<endl;

cout<<"2.Display()"<<endl;

cout<<"3.Exit()"<<endl;

cout<<"Enter your option:";

cin>>option;

switch(option)

{

case 1:

s.create();

break;

case 2:

s.display();

break;

case 3:

break;

}

}

}

1. **Write a program to implement the concept of Stack with Push, Pop, Display and Exit operations.**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

class stack

{

int stk[5];

int top;

public:

 stack()

{

 top=-1;

}

void push(int x)

{

if(top >4)

{

cout<<"stack over flow";

return;

}

stk[++top]=x;

cout<<"inserted"<<x;

}

void pop()

{

if(top <0)

{

cout<<"stack under flow";

return;

}

cout<<"deleted"<<stk[top--];

}

void display()

{

if(top<0)

{

cout<<" stack empty";

return;

}

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

cout<<stk[i]<<" ";

}

};

 main()

{

int ch;

 stack st;

while(1)

{

cout<<"\n1.push 2.pop 3.display 4.exit\nEnterur choice";

cin>>ch;

switch(ch)

{

case1:cout<<"enter the element";

cin>>ch;

st.push(ch);

break;

case2:st.pop();break;

case3:st.display();break;

case4:exit(0);

}

}

return(0);

}

**OUTPUTS**

1.push 2.pop 3.display 4.exit
Enter ur choice2
stack under flow

1.push 2.pop 3.display 4.exit
Enter ur choice1
enter the element2
inserted2

1.push 2.pop 3.display 4.exit
Enter ur choice1
enter the element3
inserted3

1.push 2.pop 3.display 4.exit
Enter ur choice2
deleted3

1.push 2.pop 3.display 4.exit
Enter ur choice1
enter the element5
inserted5

1.push 2.pop 3.display 4.exit
Enter ur choice3
5 2

1.push 2.pop 3.display 4.exit
Enter ur choice4

1. **Write a program to implement Tower of Hanoi problem.**

#include<iostream>
using namespace std;

void towers(int, char, char, char);

int main()
{
    int num;

    cout<<"Enter the number of disks : ";
    cin>>num;
    cout<<"The sequence of moves involved in the Tower of Hanoi are :\n";
    towers(num, 'A', 'C', 'B');
    return 0;
}

void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        cout<<"\n Move disk 1 from peg"<<frompeg<<"to peg "<<topeg;
        return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    cout<<"\n Move disk "<<num<<"from peg"<<frompeg<<" to peg "<<topeg;
    towers(num - 1, auxpeg, topeg, frompeg);

}

**Output:**

1. **Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

class queue

{

 int queue1[5];

 int rear,front;

 public:

 queue()

 {

 rear=-1;

 front=-1;

 }

 void insert(int x)

 {

 if(rear > 4)

 {

 cout <<"queue over flow";

 front=rear=-1;

 return;

 }

 queue1[++rear]=x;

 cout <<"inserted" <<x;

 }

 void delet()

 {

 if(front==rear)

 {

 cout <<"queue under flow";

 return;

 }

 cout <<"deleted" <<queue1[++front];

 }

 void display()

 {

 if(rear==front)

 {

 cout <<" queue empty";

 return;

 }

 for(int i=front+1;i<=rear;i++)

 cout <<queue1[i]<<" ";

 }

};

main()

{

 int ch;

 queue qu;

 while(1)

 {

 cout <<"**\n**1.insert 2.delet 3.display 4.exit**\n**Enter ur choice";

 cin >> ch;

 switch(ch)

 {

 case 1: cout <<"enter the element";

 cin >> ch;

 qu.insert(ch);

 break;

 case 2: qu.delet(); break;

 case 3: qu.display();break;

 case 4: exit(0);

 }

 }

return (0);

}

**OUTPUT**

1.insert 2.delet 3.display 4.exit
Enter ur choice1
enter the element21
inserted21

1.insert 2.delet 3.display 4.exit
Enter ur choice1
enter the element22
inserted22

1.insert 2.delet 3.display 4.exit
Enter ur choice1
enter the element16
inserted16

1. **Write a program to implement bubble sort.**

#include<iostream.h>

intmain()

{

    inta[50],n,i,j,temp;

    cout<<"Enter the size of array: ";

    cin>>n;

    cout<<"Enter the array elements: ";

        for(i=0;i<n;++i)

        cin>>a[i];

        for(i=1;i<n;++i)

    {

        for(j=0;j<(n-i);++j)

            if(a[j]>a[j+1])

            {

                temp=a[j];

                a[j]=a[j+1];

                a[j+1]=temp;

            }

    }

    cout<<"Array after bubble sort:";

    for(i=0;i<n;++i)

    cout<<" "<<a[i];

    return0;

}

**Output**

1. **Write a program to implement selection sort.**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

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

 cout<<"Enter Array Size : ";

 cin>>size;

 cout<<"Enter Array Elements : ";

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

 {

 cin>>arr[i];

 }

 cout<<"Sorting array using selection sort...\n";

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

 {

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

 {

 if(arr[i]>arr[j])

 {

 temp=arr[i];

 arr[i]=arr[j];

 arr[j]=temp;

 }

 }

 }

 cout<<"Now the Array after sorting is :\n";

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

 {

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

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **Write a program to implement insertion sort.**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

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

 cout<<"Enter Array Size : ";

 cin>>size;

 cout<<"Enter Array Elements : ";

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

 {

 cin>>arr[i];

 }

 cout<<"Sorting array using selection sort ... \n";

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

 {

 temp=arr[i];

 j=i-1;

 while((temp<arr[j]) && (j>=0))

 {

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

 j=j-1;

 }

 arr[j+1]=temp;

 }

 cout<<"Array after sorting : \n";

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

 {

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

 }

 getch();

}

When the above C++ program is compile and executed, it will produce the following result:



1. **Write a program to search the element using sequential search.**

#include<iostream>
using namespace std;
int main()

{
cout<<"Enter The Size Of Array:   ";
int size;
cin>>size;
int array[size], key,i;
for(int j=0;j<size;j++)

{
cout<<"Enter "<<j<<" Element: ";
cin>>array[j];
}
for(int a=0;a<size;a++){
   cout<<"array[ "<<a<<" ]  =  ";
   cout<<array[a]<<endl;
}
cout<<"Enter Key To Search  in Array";
cin>>key;
  for(i=0;i<size;i++)

{
   if(key==array[i])

{
cout<<"Key Found At Index Number :  "<<i<<endl;
break;
   }}
if(i != size){
cout<<"KEY FOUND at index :  "<<i;
}
else

{
cout<<"KEY NOT FOUND in Array  ";
}
   return 0;
}

**Output:**



1. **Write a program to search the element using binary search.**

#include<iostream.h>

#include<conio.h>

void main()

{

 clrscr();

 int n, i, arr[50], search, first, last, middle;

 cout<<"Enter total number of elements :";

 cin>>n;

 cout<<"Enter "<<n<<" number :";

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

 {

 cin>>arr[i];

 }

 cout<<"Enter a number to find :";

 cin>>search;

 first = 0;

 last = n-1;

 middle = (first+last)/2;

 while (first <= last)

 {

 if(arr[middle] < search)

 {

 first = middle + 1;

 }

 else if(arr[middle] == search)

 {

 cout<<search<<" found at location "<<middle+1<<"\n";

 break;

 }

 else

 {

 last = middle - 1;

 }

 middle = (first + last)/2;

 }

 if(first > last)

 {

 cout<<"Not found! "<<search<<" is not present in the list.";

 }

 getch();

}

**Output:**



