1. **Program for Bubble Sort/ascending order in C++**

#include<iostream>

using namespace std;

int main()

{

int a[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(i=0;i<n-1;i++)

{

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

if(a[j]>a[j+1]) // if(arr[j]>arr[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];

return 0;

}

## Program for Selection Sort in C++

#include<iostream>

using namespace std;

int main()

{

    int i,j,n,loc,temp,min,a[30];

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

    cin>>n;

    cout<<"\nEnter the elements\n";

    for(i=0;i<n;i++)

    {

        cin>>a[i];

    }

    for(i=0;i<n-1;i++)

    {

        min=a[i];

        loc=i;

        for(j=i+1;j<n;j++)

        {

            if(min>a[j])

            {

                min=a[j];

                loc=j;

            }

        }

        temp=a[i];

        a[i]=a[loc];

        a[loc]=temp;

    }

    cout<<"\nSorted list is as follows\n";

    for(i=0;i<n;i++)

    {

        cout<<a[i]<<" ";

    }

    return 0;

}

Or

/\* C++ Program - 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();

}

1. **Linear Search**

#include <stdio.h>

int main()

{

int array[100], search, c, n, count = 0;

printf("Enter number of elements in array\n");

scanf("%d", &n);

printf("Enter %d numbers\n", n);

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

scanf("%d", &array[c]);

printf("Enter a number to search\n");

scanf("%d", &search);

for (c = 0; c < n; c++) {

if (array[c] == search) {

printf("%d is present at location %d.\n", search, c+1);

count++;

}

}

if (count == 0)

printf("%d isn't present in the array.\n", search);

else

printf("%d is present %d times in the array.\n", search, count);

return 0;

}

1. **Binary Search**

#include <stdio.h>

int main()

{

int c, first, last, middle, n, search, array[100];

printf("Enter number of elements\n");

scanf("%d",&n);

printf("Enter %d integers\n", n);

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

scanf("%d",&array[c]);

printf("Enter value to find\n");

scanf("%d", &search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while (first <= last) {

if (array[middle] < search)

first = middle + 1;

else if (array[middle] == search) {

printf("%d found at location %d.\n", search, middle+1);

break;

}

else

last = middle - 1;

middle = (first + last)/2;

}

if (first > last)

printf("Not found! %d isn't present in the list.\n", search);

return 0;

}

## 5. ****Tower of Hanoi****

#include<stdio.h>

#include<conio.h>

int main ()

{

int n;

printf("Enter number of disks required: \n");

scanf ("%d", &n);

TOH (n, 'A', 'B',' C');

getch();

return 0;

}

void TOH (int n, char src, char spare, char dest)

{

if (n==1)

printf("Move from %c to %c \n", src, dest);

else

{ TOH(n-1, src, dest, spare) ;

TOH(1, src, spare, dest);

TOH(n-1, spare, src, dest);

}

}

1. **Knapsack**

#include <iostream>

using namespace std;

// A utility function that returns maximum of two integers

int max(int a, int b)

{

return (a > b) ? a : b;

}

// Returns the maximum value that can be put in a knapsack of capacity W

int knapSack(int W, int wt[], int val[], int n)

{

int i, w;

int K[n + 1][W + 1];

// Build table K[][] in bottom up manner

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

{

for (w = 0; w <= W; w++)

{

if (i == 0 || w == 0)

K[i][w] = 0;

else if (wt[i - 1] <= w)

K[i][w]

= max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);

else

K[i][w] = K[i - 1][w];

}

}

return K[n][W];

int main()

{

cout << "Enter the number of items in a Knapsack:";

int n, W;

cin >> n;

int val[n], wt[n];

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

{

cout << "Enter value and weight for item " << i << ":";

cin >> val[i];

cin >> wt[i]

}

// int val[] = { 60, 100, 120 };

// int wt[] = { 10, 20, 30 };

// int W = 50;

cout << "Enter the capacity of knapsack";

cin >> W;

cout << knapSack(W, wt, val, n);

return 0;

}