AIM:-

Assignment No. 04

1. Sort the data in ascending order using selection sort and descending order by using insertion sort (Display pass by pass output).
2. Search a perticular data using linear search.

OBJECTIVE:-

To understand the sorting techniques i.e. **Selection sort and Insertion sort**. To know the difference between these two sorting techniques and there inner loops i.e. sorting methods. Also to learn and to implement **Linear search** over the sorted array.

THEORY:-

Selection Sort

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1) The subarray which is already sorted.  
2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Insertion Sort

Insertion sort is the sorting mechanism where the sorted array is built having one item at a time. The array elements are compared with each other sequentially and then arranged simultaneously in some particular order. The analogy can be understood from the style we arrange a deck of cards. This sort works on the principle of inserting an element at a particular position, hence the name Insertion Sort.

Linear Search

Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

Sourcecode :-

#include<iostream>

using namespace std;

int lsearch(int a[100],int a1)

{

int i,a2,flag=0;

cout<<endl<<"Enter number you want to search :";

cin>>a2;

for(i=0;i<a1;i++)

{

if(a[i]==a2)

{

flag=1;

cout<<"Number is found and index is : "<<i<<endl;

}

}

if(flag == 0)

{

cout<<"Number is not found\n";

}

}

int isort(int a[100],int a1)

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*INSERTION SORT\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,j,k,l;

for(i=0;i<a1;i++)

{

k=a[i];

j=i-1;

while(j>=0 && a[j]<k)

{

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

j--;

}

a[j+1]=k;

cout<<endl;

for(l=0;l<a1;l++)

{

cout<<a[l]<<" ";

}

}

cout<<endl<<"\nDescending order is : ";

for(i=0;i<a1;i++)

{

cout<<a[i]<<" ";

}

}

int ssort(int a[100],int a1)

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*SELECTION SORT\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,j,k,temp;

for(i=0;i<a1-1;i++)

{

for(j=i+1;j<a1;j++)

{

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

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

cout<<endl;

for(k=0;k<a1;k++)

{

cout<<a[k]<<" ";

}

}

}

}

cout<<endl<<"\nAscending order is: ";

for(i=0;i<a1;i++)

{

cout<<a[i]<<" ";

}

}

int main()

{

int a[100],a1,i,a3;

cout<<"How many number you have : ";

cin>>a1;

cout<<"Enter element\n";

for(i=0;i<a1;i++)

{

cin>>a[i];

}

cout<<"\nYour list is : ";

for(i=0;i<a1;i++)

{

cout<<a[i]<<" ";

}

cout<<"\n";

ssort(a,a1);

cout<<"\n";

isort(a,a1);

lsearch(a,a1);

return 0;

}

Output:-

/\*

How many number you have : 5

Enter element

12

4

33

5

17

Your list is : 12 4 33 5 17

\*\*\*\*\*\*\*\*\*\*\*SELECTION SORT\*\*\*\*\*\*\*\*\*\*\*\*\*

4 12 33 5 17

4 5 33 12 17

4 5 12 33 17

4 5 12 17 33

Ascending order is: 4 5 12 17 33

\*\*\*\*\*\*\*\*\*\*\*INSERTION SORT\*\*\*\*\*\*\*\*\*\*\*\*\*

4 5 12 17 33

5 4 12 17 33

12 5 4 17 33

17 12 5 4 33

33 17 12 5 4

Descending order is : 33 17 12 5 4

Enter number you want to search :12

Number is found and index is : 2

--------------------------------

Process exited after 43.26 seconds with return value 0

Press any key to continue . . .

\*/

Conclusion:-

Understood the sorting techniques i.e. **Selection sort and Insertion sort**. Also

the difference between these two sorting techniques and there inner loops i.e. sorting

methods.

1. The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning.
2. Insertion sort is the sorting mechanism where the sorted array is built having one item at a time.
3. In linear search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned