1. 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.

Algorithm :-

ALGORITHM FOR SELECTION SORT

1. Set the first element as minimum.
2. Compare minimum with the second element. If the second element is smaller than , minimum assign second element as minimum. Compare minimum with the third element. Again, if the third element is smaller, then assign minimum to the third element otherwise do nothing. The process goes on until the last element.
3. After each iteration , minimum is placed in the front of the unsorted list.
4. For each iteration, indexing starts from the first unsorted element. Step 1 to 3 are repeated until all the elements are placed at their correct positions.

ALGORITHM FOR INSERTION SORT

1. The first element in the array is assumed to be sorted. Take the second element and store it separately in key.  
     
   Compare key with the first element. If the first element is greater than key , then key is placed in front of the first element.
2. Now, the first two elements are sorted.  
     
   Take the third element and compare it with the elements on the left of it. Placed it just behind the element smaller than it. If there is no element smaller than it, then place it at the beginning of the array.
3. In a similar way, place every unsorted element at its correct position.

ALGORITHM FOR LINEAR SEARCH

1. Here in this program we have taken the array as an input from the user along with the key to be searched. We have created a separate function for Linear Search.  
2. When the function is called we have to run a loop n times where n is the number of elements in an array.  
3. In each iteration we are comparing the value of key with elements of array in increasing order of array index.  
4. If key is equal to one of the array elements we print the value of index at which we found them to be equal.

SOURCE CODE:

/\*

PROG BY ANOM DEVGUN

GR: 21810017

\*/

#include<iostream>

using namespace std;

void dispa(int A[],int n)

{

int l=0;

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

{

cout<<A[l]<<" ";

}

}

int main()

{

char ch;

int n,i=0,j=0,ke=0,k=0;

int temp=0,min=0;

cout<<"Enter the number of elements you want in array:\n";

cin>>n;

int a[n];

cout<<"Enter "<<n<<" numbers\n";

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

{

cin>>a[i];

}

cout<<"1)For selection sorting the input data:\n";

cout<<"2)For insertion sorting the input data:\n";

cout<<">>";

cin>>ch;

switch (ch)

{

case '1':

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

{

min=i;

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

{

if(a[j]<a[min])

{

Min = j;

}

}

temp=a[min];

a[min]=a[i];

a[i]=temp;

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

{

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

}

cout<<"\n";

}

break;

case '2':

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

{

ke=a[i];

j=i-1;

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

{

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

j=j-1;

}

a[j+1]=ke;

dispa(a,i+1);

cout<<"\n";

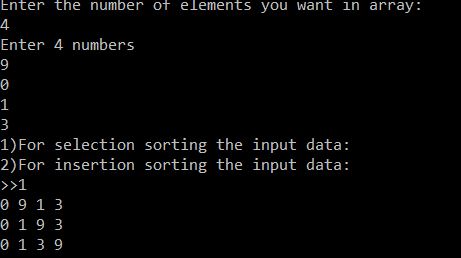
}

break;

}

return 0;

}

****