**Debre Markos University**

**Department of Information Technology**

**Data Structure and Algorithm Lab Manual**

**Lab 2: Insertion, Selection and Bubble sort implementation.**

**Tools used:** Quincy 2005v.1.3 editor

**Objective:**

Students should be able to know:

- Differentiate different sorting algorithms.

- Identify their role and when and how do we implement them

**//Insertion Sort**

#include<iostream.h>

int main()

{

int list[10];

int temp,n,i,m;

cout<<"Enter the size of array [Upto 10]: ";

cin>>n;

cout<<"Enter the array elements: "<<endl;

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

{

cout<<"\tEnter element "<<(m+1)<<" : ";

cin>>list[m];

}

cout<<"\t\tThe array before sorting is : { ";

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

{

cout<<list[m]<<" ";

}

cout<<"}"<<endl;

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

{

temp=list[i];

for(int j=i;j>0&&temp<list[j-1];j--)

{

list[j]=list[j-1];

list[j-1]=temp;

}

}

cout<<"\t\tThe array after sorting is: { ";

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

{

cout<<list[k]<<" ";

}

cout<<"}";

return 0;

}

**1**

**//Selection Sort**

#include<iostream.h>

int main()

{

int i,j,n,m,k,temp,smallest;

int list[10];

cout<<"Enter the size of array [Up to 10]: ";

cin>>n;

cout<<"Enter elements of the array: "<<endl;

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

{

cout<<"\t Enter element "<<(m+1)<<" : ";

cin>>list[m];

}

cout<<"\t\t The array before sorting is : { ";

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

{

cout<<list[m]<<" ";

}

cout<<"}"<<endl;

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

{

smallest=i;

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

{

if(list[j]<list[smallest])

smallest=j;

}

temp=list[smallest];

list[smallest]=list[i];

list[i]=temp;

}

cout<<"\t\t The array after sorting is: { ";

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

{

cout<<list[k]<<" ";

}

cout<<"}";

return 0;

}

**//Bubble sort**

#include<iostream.h>

int main()

{

int i,j,m;

int list[10];

int temp,n;

cout<<"Enter the size of array [Up to 10]: ";

cin>>n;

cout<<"Enter the array elements: "<<endl;

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

{

cout<<"\t Enter element "<<(m+1)<<" : ";

cin>>list[m];

}

cout<<"\t\t The array before sorting is : { ";

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

cout<<list[m]<<" ";

cout<<"}"<<endl;

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

{

for(j=n-1;j>i; j--)

{

if(list[j]<list[j-1])

{

temp=list[j];

list[j]=list[j-1];

list[j-1]=temp;

}

}

}

cout<<"\t\t Array after sorting is: { ";

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

{

cout<<list[k]<<" ";

}

cout<<"}";

return 0;

}

**Selection, Insertion and Bubble Sort**

Sorting — arranging items in order — is the most fundamental task in computation. Sorting enables efficient searching algorithms such as binary search.

Selection, insertion and bubble sort are easily understandable and also similar to each other, but they are less efficient than merge sort or quick sort. The basic ideas are as below:

**Selection sort**: repeatedly pick the smallest element to append to the result.

*Selection sort* is to repetitively pick up the smallest element and put it into the right position:

* Find the smallest element, and put it to the first position.
* Find the next smallest element, and put it to the second position.
* Repeat until all elements are in the right positions.

A loop through the array finds the smallest element easily. After the smallest element is put in the first position, it is fixed and then we can deal with the rest of the array.

**Insertion sort**: repeatedly add new element to the sorted result.

*Insertion sort* maintains a sorted sub-array, and repetitively inserts new elements into it. The process is as following:

* Take the first element as a sorted sub-array.
* Insert the second element into the sorted sub-array (shift elements if needed).
* Insert the third element into the sorted sub-array.
* Repeat until all elements are inserted.

**Bubble sort**: repeatedly compare neighbor pairs and swap if necessary.

Bubble sort repetitively compares adjacent pairs of elements and swaps if necessary.

* Scan the array, swapping adjacent pair of elements if they are not in relative order. This bubbles up the largest element to the end.
* Scan the array again, bubbling up the second largest element.
* Repeat until all elements are in order.

**2**