Mechi Multiple Campus

(Tribhuvan University)
Bhadrapur, Jhapa

Lab Report of

Data Structures and Algorithm (CACS-201)

Implementation of Sorting Algorithm

Faculty of Humanities & Social Sciences
Tribhuvan University
Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus
Department of Bachelor in Computer Application
Bhadrapur, Jhapa, Nepal

Introduction to Sorting and Sorting Algorithms

Sorting is the process of arranging data in some logical order such as ascending and descending. Sorting is important as it support for faster searching of data.

Sorting can be:-

Internal Sorting:- It refers to the process of sorting data from the internal memory. It is faster but applicable for limited number of data.

External Sorting:- It refers to the process of sorting data from the external file by reading from the external memory. It is slower than the internal sorting but is applicable for large number of data.

Sorting Algorithms

Some of the common Sorting algorithms are:-

- Bubble Sort
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Radix Sort
- Shell Sort
- Exchange Sort
- Binary Sort

Algorithm, Program Code and Output

Bubble Sort

Algorithm

(1) START

(2) For the first iteration, Compare all the elements (N). For the subsequent runs, compare (N-1), N-2) and so on.

(3) Compare each elements with its right neighbour

(4) Swap the smallest element to the left

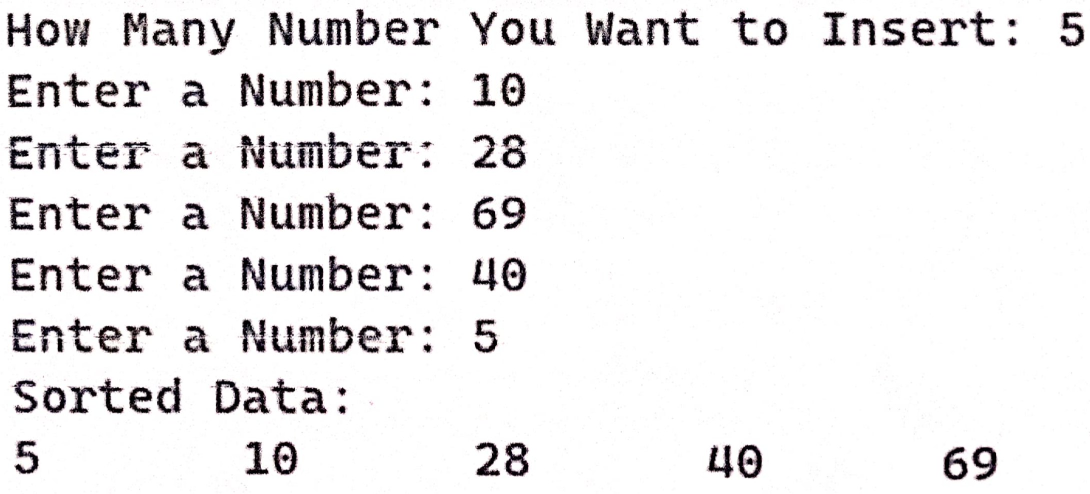
(5) Keep repeating steps 2 to 4 until the Whole list is covered

(6) END

Program Code

```
#include<stdio.h>
void main(){
   int n,num[100],i,j,temp;
   printf("How Many Number You Want to Insert: ");
   scanf("%d",&n);
   for(i=0;i<n;i++){
         printf("Enter a Number: ");
         scanf("%d",&num[i]);
         }
         for(i=0;i<n;i++){
                 for(j=0;j<n;j++){
                         if(num[j]>num[j+1]){
                                 temp=num[j];
                                 num[j]=num[j+1];
                                 num[j+1]=temp;
                         }
                 }
         }
         printf("Sorted Data: \n");
         for(i=0;i<n;i++){
                 printf("%d\t",num[i]);
         }
}
```

Output of the Program



```
How Many Number You Want to Insert: 5
Enter a Number: 10
Enter a Number: 28
Enter a Number: 69
Enter a Number: 40
Enter a Number: 5
Sorted Data:
5       10      28      40      69
```

Selection Sort

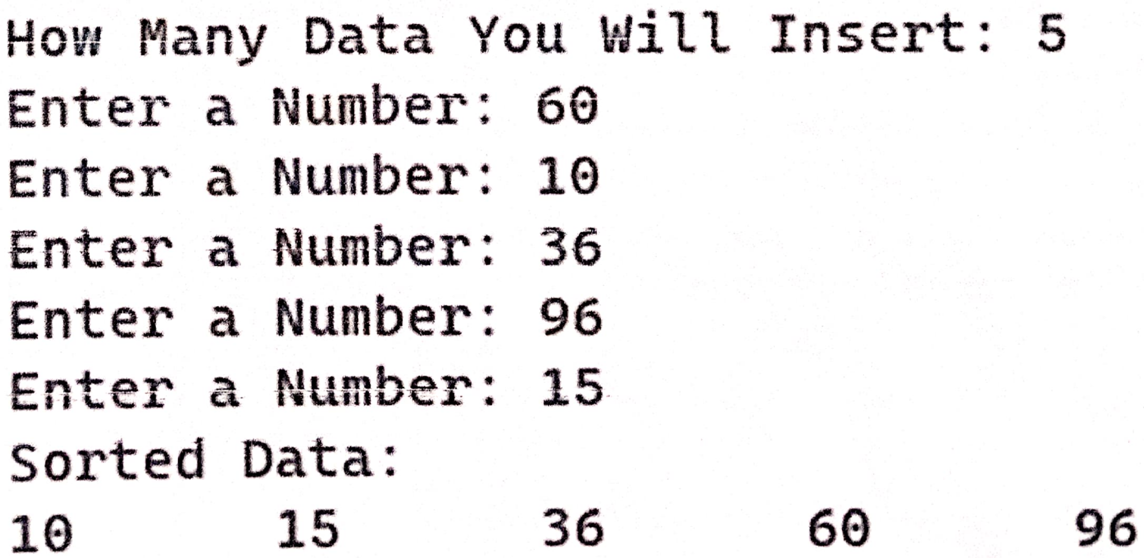
Algorithm

```
① START
② Repeat step 2 to 5 for i=0,1,2,---N-2
③ Min = A[i], loc=i
④ Repeat step 5 for j=i+1, i+2, ---- N-1
⑤ If A[j] < min, then:
        Min = A[j] and loc = j
⑥ Interchange A[i] and A[LOC]
        Temp = A[i]
        A[i] = A[loc]
        A[loc] = Temp
⑦ EXIT
```

Program Code

```
#include<stdio.h>
void main(){
        int n,num[100],i,j,temp,min,loc;
        printf("How Many Data You Will Insert: ");
        scanf("%d",&n);
        for(i=0;i<n;i++){
                printf("Enter a Number: ");
                scanf("%d",&num[i]);
        }
        for(i=0;i<n;i++){
                min=num[i];
                loc=i;
                for(j=i+1;j<n;j++){
                        if(num[j]<min){
                                min=num[j];
                                loc=j;
                        }
                }
                temp=num[i];
                num[i]=num[loc];
                num[loc]=temp;
        }
        printf("Sorted Data: \n");
        for(i=0;i<n;i++){
                printf("%d\t",num[i]);
        }
  }
```

Output of the Program



```
How Many Data You Will Insert: 5
Enter a Number: 60
Enter a Number: 10
Enter a Number: 36
Enter a Number: 96
Enter a Number: 15
Sorted Data:
10      15      36      60      96
```

Insertion Sort

Algorithm

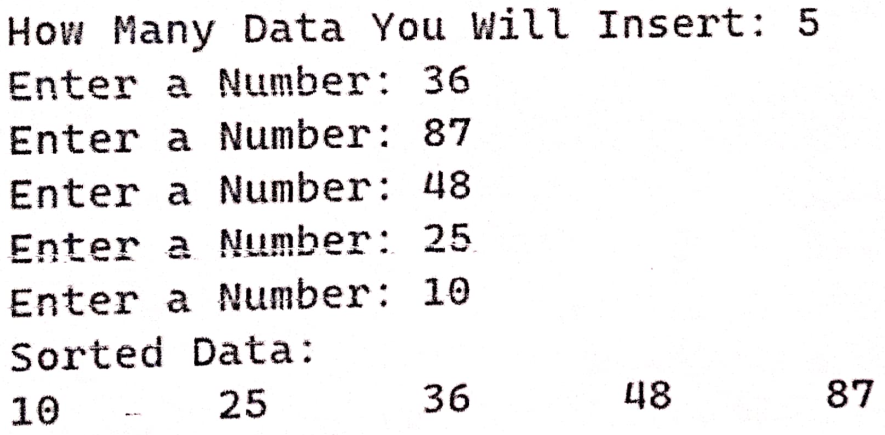
The following algorithm sorts N Data in the array:-

```
① START
② Repeat step 3 to 5 for i = 0, 1, 2 --- N-1
③ Temp = A[i] and PTR = i-1
④ Repeat While temp < A[PTR]
        a) Set A[PTR+1] = A[PTR]
        b) Set PTR = PTR-1
⑤ Set A[PTR+1] = temp
⑥ EXIT
```

Program Code

```
#include<stdio.h>
void main(){
        int num[100],n,ptr,i,j,temp;
        printf("How Many Data You Will Insert: ");
        scanf("%d",&n);
        for(i=0;i<n;i++){
                printf("Enter a Number: ");
                scanf("%d",&num[i]);
        }
        for(i=0;i<n;i++){
                temp=num[i];
                ptr=i-1;
                while(temp<num[ptr]&&ptr>=0){
                        num[ptr+1]=num[ptr];
                        ~~ptr--;~~
                }
                num[ptr+1]=temp;
        }
                printf("Sorted Data: \n");
        for(i=0;i<n;i++){
                ~~printf("%d\t",num[i]);~~
        }
}
```

Output of the Program



```
How Many Data You Will Insert: 5
Enter a Number: 36
Enter a Number: 87
Enter a Number: 48
Enter a Number: 25
Enter a Number: 10
Sorted Data:
10      25      36      48      87
```

Conclusion

In Sorting, the data are arranged in some logical order like ascending order or descending order. By performing the sorting in data, Searching of data will be faster.