Batch: C2-1 Roll No.: 16010122104

Experiment / assignment / tutorial No. 5

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Program to sort array

AIM: Program to sort the 1D array in the ascending or descending order and then accept the element from user and insert in the same array at its correct place by keeping array sorted

---

Expected OUTCOME of Experiment:

To successfully run a array program, where you take a user defined array and sort it.

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving, G. Michael Schneider ,Wiley India edition.
4. **http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**

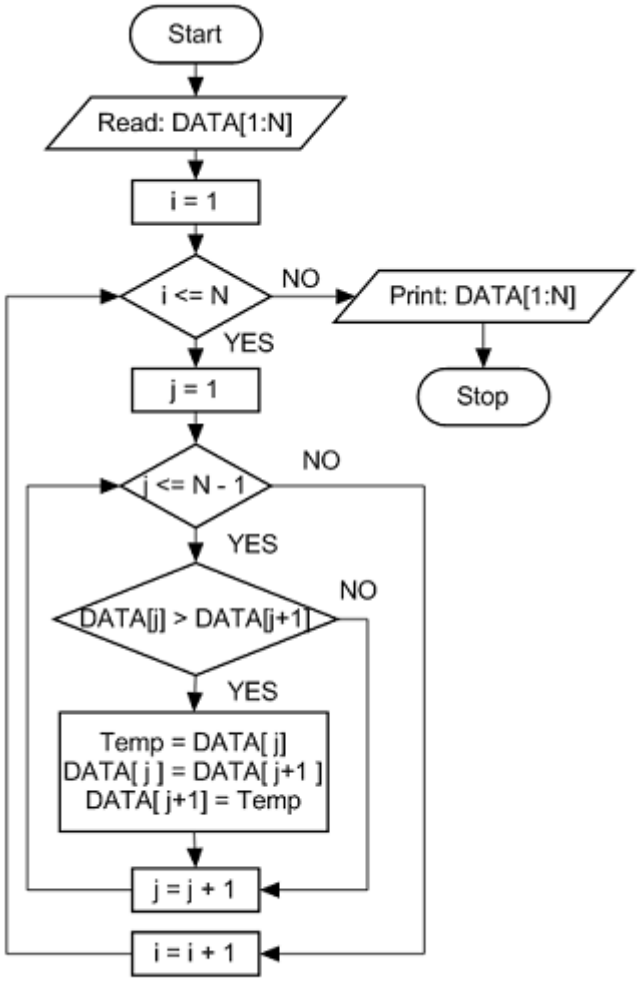
Problem Definition:

The program takes a 1D array and sorts it in the specified manner. The user enters an element and the same has to be inserted at the correct place in the sorted array.

```
#include<stdio.h>
void main()
{
   int i, j;
   printf("Enter the size of array:");
   int n;
   scanf("%d",&n);
   int arr[n];
   printf("Enter the elements of the array:\n");
   for(int i=0; i<n; i++)
      {
         scanf("%d", &arr[i]);
      }
   for(i=0; i<n; i++)
   {
      for(j=i+1; j<n;j++)
      {
         if(arr[i]>arr[j])
         {
            int temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
         }
      }
   }
```

```
   printf("Sorted array is: \n");
   for(int i=0; i<n; i++)
   {
      printf("%d\n",arr[i]);
   }
}
```

Flowchart:



Bubble Sort Flowchart

Implementation details:

Input: *arr[] = {5, 1, 4, 2, 8}*

First Pass:

- *Bubble sort starts with very first two elements, comparing them to check which one is greater.*
- *(**5 1** 4 2 8) –> (**1 5** 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1.*
- *(1 **5 4** 2 8) –> (1 **4 5** 2 8), Swap since 5 > 4*
- *(1 4 **5 2** 8) –> (1 4 **2 5** 8), Swap since 5 > 2*
- *(1 4 2 **5 8**) –> (1 4 2 **5 8**), Now, since these elements are already in order (8 > 5), algorithm does not swap them.*

Second Pass:

- *Now, during second iteration it should look like this:*
- *(**1 4** 2 5 8) –> (**1 4** 2 5 8)*
- *(1 **4 2** 5 8) –> (1 **2 4** 5 8), Swap since 4 > 2*
- *(1 2 **4 5** 8) –> (1 2 **4 5** 8)*
- *(1 2 4 **5 8**) –> (1 2 4 **5 8**)*

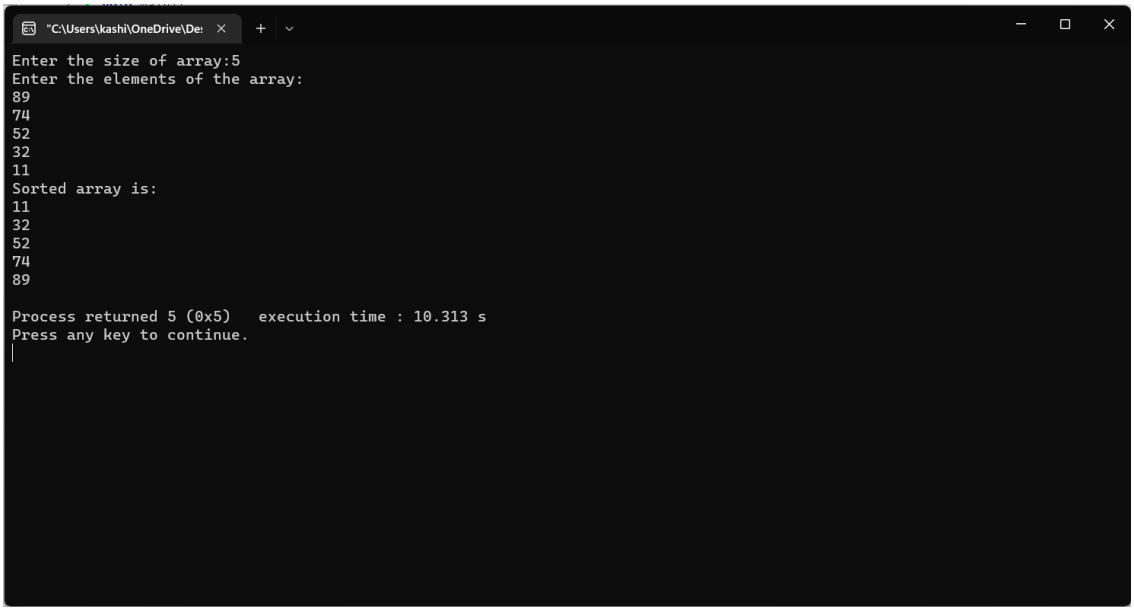
Third Pass:

- *Now, the array is already sorted, but our algorithm does not know if it is completed.*
- *The algorithm needs one **whole** pass without **any** swap to know it is sorted.*
- *(**1 2** 4 5 8) –> (**1 2** 4 5 8)*
- *(1 **2 4** 5 8) –> (1 **2 4** 5 8)*
- *(1 2 **4 5** 8) –> (1 2 **4 5** 8)*
- *(1 2 4 **5 8**) –> (1 2 4 **5 8**)*

Illustration:

		0	1	2	3	4	5	6	7
i = 0	j								
	0	5	3	1	9	8	2	4	7
	1	3	5	1	9	8	2	4	7
	2	3	1	5	9	8	2	4	7
	3	3	1	5	9	8	2	4	7
	4	3	1	5	8	9	2	4	7
	5	3	1	5	8	2	9	4	7
	6	3	1	5	8	2	4	9	7
i=1	0	3	1	5	8	2	4	7	9
	1	1	3	5	8	2	4	7	
	2	1	3	5	8	2	4	7	
	3	1	3	5	8	2	4	7	
	4	1	3	5	2	8	4	7	
	5	1	3	5	2	4	8	7	
i=2	0	1	3	5	2	4	7	8	
	1	1	3	5	2	4	7		
	2	1	3	5	2	4	7		
	3	1	3	2	5	4	7		
	4	1	3	2	4	5	7		
i=3	0	1	3	2	4	5	7		
	1	1	3	2	4	5			
	2	1	2	3	4	5			
	3	1	2	3	4	5			
i=4	0	1	2	3	4	5			
	1	1	2	3	4				
	2	1	2	3	4				
i=5	0	1	2	3	4				
	1	1	2	3					
i=6	0	1	2	3					
		1	2						

Output(s):



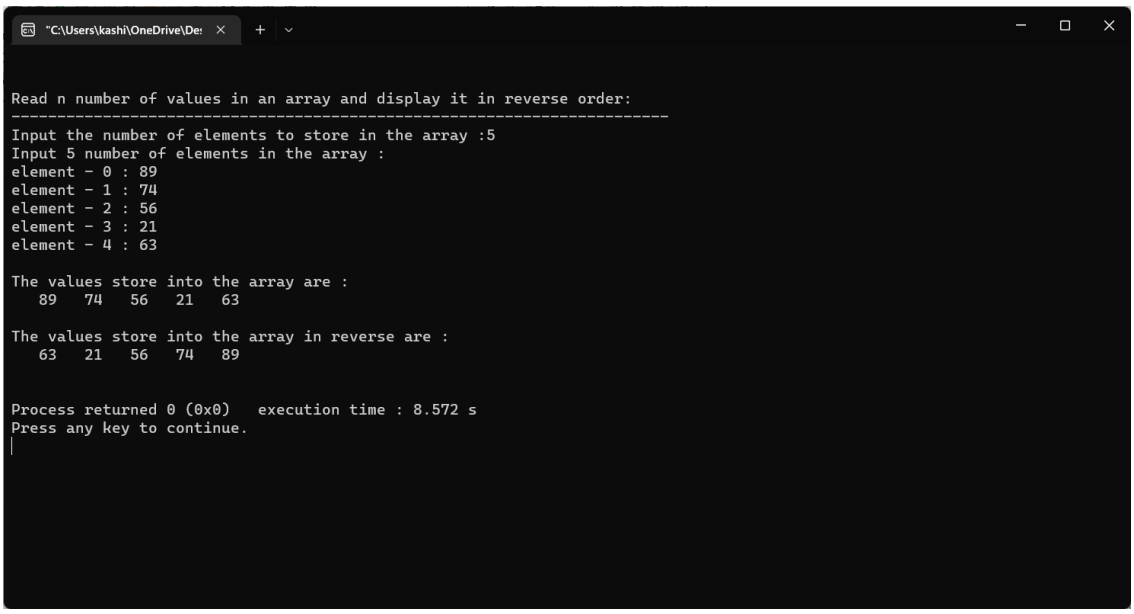
Conclusion:

Numbers of the array are rightly arranged in increasing or decreasing order , & the input value has been placed rightly in given order.

Post Lab Descriptive Questions

Write a program to enter n numbers, store them in an array and rearrange the array in the reverse order.

```
#include <stdio.h>

void main()
{
  int i,n,a[100];

      printf("\n\nRead n number of values in an array and display it in reverse order:\n");
      printf("------------------------------------------------------------------------\n");

  printf("Input the number of elements to store in the array :");
  scanf("%d",&n);

  printf("Input %d number of elements in the array :\n",n);
  for(i=0;i<n;i++)
    {
 printf("element - %d : ",i);
 scanf("%d",&a[i]);
 }

  printf("\nThe values store into the array are : \n");
  for(i=0;i<n;i++)
    {
  printf("% 5d",a[i]);
 }

  printf("\n\nThe values store into the array in reverse are :\n");
  for(i=n-1;i>=0;i--)
    {
  printf("% 5d",a[i]);
 }
  printf("\n\n");
}
```



Date: 07/01/2023 **Signature of faculty in-charge**