**Batch: B2 Roll No.: 121**

**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:** Write a C 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:**

**CO3:** Illustrate the use of derived and structured data types such as arrays, strings, structures and unions.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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/**](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.

**Flowchart:**

**Start**

Declare num[100], n, i, j, temp, ins\_num, pos\_num, new\_num[100]

Read n

Input n elements of number array

Display: The array sorted in descending order is as follows

Is i < n?

Is j < n?

Is num[i] < num[j]?

**True False**

temp = num[i]

num[i] = num[j]

num[j] = temp

Display num[i]

Display: The array sorted in ascending order is as follows

Is i < n?

Is j < n?

Is num[i] > num[j]?

temp = num[i]

num[i] = num[j]

num[j] = temp

Display num[i]

Read ins\_num

Is i < n?

Is ins\_num < new\_num[i]?

pos\_num = i + 1

pos\_num = i

Is i >= pos\_num?

num[i + 1] = num[i]

Display new\_num[100] array

Stop

**Implementation details:**

#include<stdio.h>

int main()

{

int num[100], n, i, j, temp;

int ins\_num, pos\_num, new\_num[100];

printf(“\nEnter the number of elements: “);

scanf(“%d”, &n);

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

{

printf(“\nEnter number %d: “, i+1);

scanf(“%d”, &num[i]);

}

printf(“\nThe array sorted in the descending order is as follows:”);

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

{

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

{

if(num[i]<num[j])

{

temp = num[i];

num[i] = num[j];

num[j] = temp;

}

}

printf(“ %d”, num[i]);

}

printf(“\nThe array sorted in the ascending order is as follows:”);

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

{

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

{

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

{

temp = num[i];

num[i] = num[j];

num[j] = temp;

}

}

printf(“ %d”, num[i]);

}

//now, the elements of first array, which is sorted in the ascending order,

//will be transferred, in the same order, to a new array so that the element

//insertion operation occurs seamlessly.

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

{

new\_num[i] = num[i];

}

printf(“\nEnter the number to be added to the array: “);

scanf(“%d”, &ins\_num);

//determining the correct position of the entered number

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

{

if(ins\_num < new\_num[i])

{

pos\_num = i;

break;

}

else

{

pos\_num = i + 1;

}

}

//shifting the elements greater than the entered element to the right

//so as to make space for the entered element.

for(i = (n+1); i >= pos\_num; i--)

{

new\_num[i+1] = new\_num[i];

}

//insert entered number in the correct place

new\_num[pos\_num] = ins\_num;

//print the modified array

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

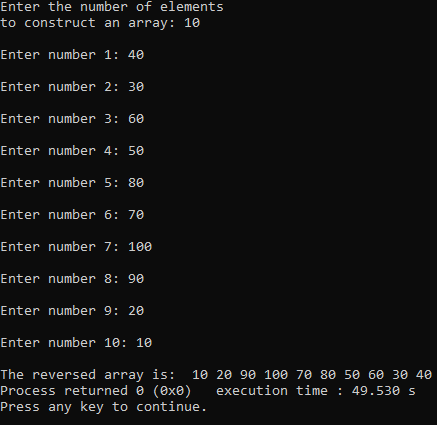
{

printf(“ %d”, new\_num[i]);

}

}

**Output(s):**



**Conclusion:**

Thus, the use of structured data type, which is, array, has been illustrated. In this experiment, an array has been used to store a number of elements of the same data type. Some operations were performed on the array which included sorting the array in descending and ascending order and to insert an element in ascending-order sorted array without disturbing the order. Thus, it can be concluded that operations such as sorting and insertion can be performed on an array.

**Post Lab Descriptive Questions**

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

#include<stdio.h>

int main()

{

int i, n, num[100], temp\_num[100];

printf(“Enter the number of elements\nto construct an array: “);

scanf(“%d”, &n);

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

{

printf(“\nEnter number %d: “, i+1);

scanf(“%d”, &num[i]);

}

//now, in the next for loop, the array will be reversed and stored in a temporary array

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

{

temp\_num[i] = num[n-i-1];

}

//now, the elements of the temporary array will be sent in the proper order to the original array

printf(“\nThe reversed array is: “);

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

{

num[i] = temp\_num[i];

printf(“ %d”, num[i]);

}

return 0;

}

**Date: \_\_12-12-2021\_\_ Signature of faculty in-charge**