Batch: P4-1 **Roll No.:** **16010423076**

Experiment / assignment / tutorial No. 4

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

Signature of the Staff In-charge with date

TITLE: Write a program in C to demonstrate use of arrays

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
Write a program to find the Transpose of a Matrix.

Expected OUTCOME of Experiment:
Apply basic concepts of C programming for problem solving.(CO1 and CO2).

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:

1. 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.

2. Write a program to find the Transpose of a Matrix.

- Entered matrix:

```
1  4  0

-5  2  7
```

- Transpose of the matrix:

```
1    -5
4    2

0    7
```

Algorithm:

1)

Ask the user for the size of the array (n).

Create an array num of size n to store the elements.

Use a loop to get user input for each element.

Use nested loops to compare and swap elements, sorting the array in ascending order.

The outer loop (i) goes through each element.

The inner loop (j) compares and swaps elements if needed.

Print the sorted array in ascending order.

Ask the user for an element to be inserted (inp).

Initialize pos to 0, the correct position for the new element.

Use a loop to find the correct position (pos) for the new element.

Increment pos until an element greater than or equal to inp is found.

Use a loop to shift elements to make space for the new element.

Start from the end and move towards the correct position.

Shift each element one position to the right.

Insert the input element at the correct position (pos) in the array.

Print the modified array, including the newly inserted element.

2)

Declare and initialize a matrix mat of size m x n.

Print the original matrix.

For each column i from 0 to n-1:

a. For each row j from 0 to m-1:

- Print mat[j][i] to obtain the transpose.

Print the transpose matrix.

End of Program.

Implementation details:

1)

```
#include<stdio.h>
void main(){
int n,i,j,a,inp;
int pos = 0;

printf("Enter the size of the Array: ");
scanf("%d", &n);
int num[n];

for(i=0;i<n;i++){
printf("Enter The Number : ");
scanf("%d", &num[i]);
}

for (i=0;i<n;i++)
{
   for (j=i+1;j<n;j++)
   {
   if (num[i] > num[j])
      {
      a =  num[i];
      num[i] = num[j];
      num[j] = a;
      }

   }

}

printf("Array in ascending order : ");
for(i=0;i<n;i++){
printf("%d\n", num[i]);
}

printf("Enter the element to be entered : ");
scanf("%d",&inp);

for(i=0;i<n;i++){
   if(num[i]<inp){
      pos++;
```

```
    }else{
       break;
    }
}

for(i=n-1;i>=pos;i--) {
num[i+1]=num[i];
}

num[pos] = inp;

printf("\nThe new array is : ");
for(i=0;i<=n;i++) {
printf("%d",num[i]);
}

}
```

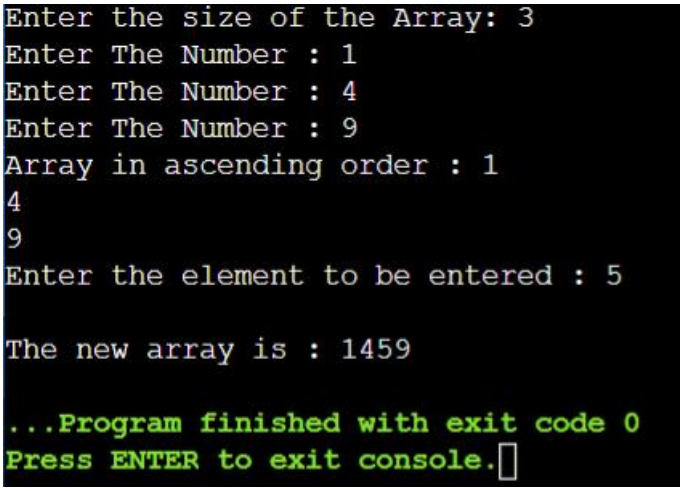
2)

```
#include<stdio.h>

void main() {
   int i,j;
   int mat[2][3] = {{1,4,0},
                    {-5,2,7}};

   printf("Matrix:\n");

for(i=0;i<2;i++) {
   for(j=0;j<3;j++) {
   printf("%d ",mat[i][j]);
   }
   printf("\n");
}

printf("\nTranspose Matrix:\n");

for(i=0;i<3;i++) {
   for(j=0;j<2;j++) {
   printf("%d ",mat[j][i]);
```

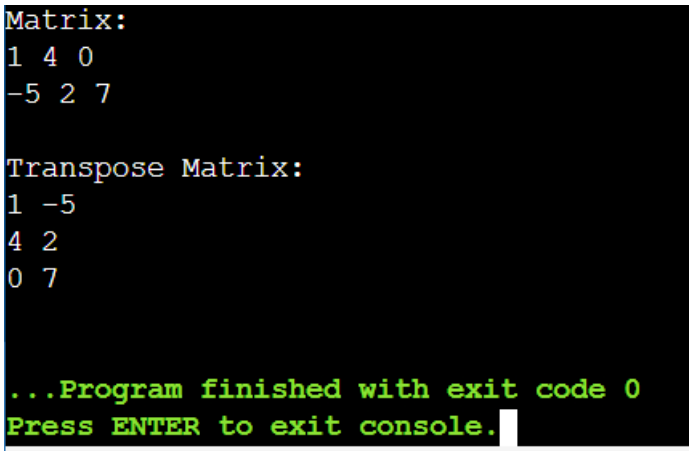
```
    }
    printf("\n");
}

}
```

Output(s):

1)


Enter the size of the Array: 3
Enter The Number : 1
Enter The Number : 4
Enter The Number : 9
Array in ascending order : 1
4
9
Enter the element to be entered : 5
The new array is : 1459
...Program finished with exit code 0
Press ENTER to exit console.

2)


Matrix:
1 4 0
-5 2 7
Transpose Matrix:
1 -5
4 2
0 7
...Program finished with exit code 0
Press ENTER to exit console.

Conclusion:

In the given C programs, I learned the fundamentals of array manipulation, sorting, and matrix operations. The first program focuses on sorting a 1D array in ascending or descending order and then efficiently inserting an element at its correct position, ensuring the array remains sorted. The second program calculates the transpose of a 2x3 matrix, demonstrating the use of nested loops for matrix manipulation.

Post Lab Questions

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

Input:

```
#include<stdio.h>
void main(){
int n,i;

printf("Enter the size of the Array: ");
scanf("%d", &n);
int num[n];

for(i=0;i<n;i++){
printf("Enter The Number : ");
scanf("%d", &num[i]);
}

for(i=n-1;i>=0;i--){
printf("\n%d", num[i]);
}
}
```

Output:

```
Enter the size of the Array: 3
Enter The Number : 13
Enter The Number : 18
Enter The Number : 19

19
18
13

...Program finished with exit code 0
Press ENTER to exit console.
```

2. Write a program which performs the following tasks:

a) Initialize an integer array of 10 elements in main( )

b) Pass the entire array to a function modify( )

c) In modify( ) multiply each element of array by 3

d) Return the control to main( ) and print the new array elements in main( )

Input:

```
#include <stdio.h>
void modify(int myarr[]){
   for(int i=0;i<10;i++){
      myarr[i] = myarr[i] * 3;
   }
}

void main()
{
int myarr[10];
printf("Enter 10 numbers separated by spaces : ");
for(int i=0;i<10;i++){
   scanf("%d",&myarr[i]);
}

printf("The original array is : ");
for(int i=0;i<10;i++){
   printf(" %d",myarr[i]);
}
```

```
modify(myarr);

printf("\nThe 3X array is : ");
for(int i=0;i<10;i++){
   printf(" %d",myarr[i]);
}

}
```

Output:

```
Enter 10 numbers separated by spaces : 1 2 3 4 5 6 7 8 9 10
The original array is :  1 2 3 4 5 6 7 8 9 10
The 3X array is :  3 6 9 12 15 18 21 24 27 30

...Program finished with exit code 0
Press ENTER to exit console.
```

Date: _____ **Signature of faculty in-charge**