

Batch: C1-2 Roll No.: 16010123036

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)

1. Start

2. Declare an integer variable n.

3. Prompt the user to enter the size of the array.

4. Read and store the entered size in the variable n.

5. Declare an integer array arr of size n.

6. Prompt the user to enter n elements for the array.

7. Use a loop to read and store each element in the array.

8. Implement the Bubble Sort algorithm to sort the array in ascending order:

a. Use nested loops for iteration.

b. Compare adjacent elements and swap them if they are in the wrong order.

9. Declare an integer variable element.

10. Prompt the user to enter the element to be inserted.

11. Read and store the entered element in the variable element.

12. Iterate over the sorted array to find the correct position for the new element:
 - a. Use a loop starting from the end of the array.
 - b. Shift elements to the right until the correct position for the new element is found.
 13. Insert the element at the correct position in the array and update the size of the array.
 14. Print "Sorted array with inserted element:".
 15. Use a loop to display the sorted array with the inserted element.
 16. End
- 2)
1. Start
 2. Declare a 3x3 integer matrix named 'matrix'.
 3. Print "Enter the 3x3 matrix:".
 4. Use nested loops to read and store values in the matrix from user input.
 - a. Outer loop (i) for rows (0 to 2).
 - b. Inner loop (j) for columns (0 to 2).
 - c. Read and store the entered integer in matrix[i][j].
 5. Print "Entered matrix:".
 6. Use nested loops to display the entered matrix.
 - a. Outer loop (i) for rows (0 to 2).
 - b. Inner loop (j) for columns (0 to 2).
 - c. Print matrix[i][j] followed by a tab.
 - d. Move to the next line after each row is printed.
 7. Print "Transpose of the matrix:".
 8. Use nested loops to calculate and display the transpose of the matrix.
 - a. Outer loop (i) for rows (0 to 2).

- b. Inner loop (j) for columns (0 to 2).
 - c. Print matrix[j][i] followed by a tab to get the transpose.
 - d. Move to the next line after each row of the transpose is printed.
9. End

Implementation details:

```
#include <stdio.h>
int main() {
    int n;
    printf("Amandeep Singh\n16010123036\n");
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    int element;
    printf("Enter the element to be inserted: ");
    scanf("%d", &element);
    int i;
    for (i = n - 1; i >= 0 && arr[i] > element; i--) {
        arr[i + 1] = arr[i];
    }
    arr[i + 1] = element;
    n++;
    printf("Sorted array with inserted element: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

2)

```
#include <stdio.h>
int main() {
    int matrix[3][3];
    printf("Amandeep Singh\n16010123036\n");
    printf("Enter the 3x3 matrix:\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("Entered matrix:\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
    printf("Transpose of the matrix:\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            printf("%d\t", matrix[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

Output(s):

1)

```
Amandeep Singh
16010123036
Enter the size of the array: 5
Enter 5 elements:
1 3 4 5 6
Enter the element to be inserted: 2
Sorted array with inserted element: 1 2 3 4 5 6
```

2)

```
Amandeep Singh
16010123036
Enter the 3x3 matrix:
1 2 3
4 5 6
7 8 9
Entered matrix:
1      2      3
4      5      6
7      8      9
Transpose of the matrix:
1      4      7
2      5      8
3      6      9
```

Conclusion:

This experiment showcased fundamental C programming concepts by performing tasks such as sorting arrays, inserting elements into sorted arrays, and computing matrix transposes. Through these exercises, participants enhanced their understanding of core programming principles and problem-solving techniques in C.

Post Lab Questions

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

Sol:


```
#include <stdio.h>
int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d numbers:\n", n);
    for (int i = 0; i < n; ++i) {
        scanf("%d", &arr[i]);
    }
    printf("Original array: ");
    for (int i = 0; i < n; ++i) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    int temp;
    for (int i = 0; i < n / 2; ++i) {
        temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }
    printf("Reversed array: ");
    for (int i = 0; i < n; ++i) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

```
Enter the number of elements: 5
Enter 5 numbers:
1 3 5 7 9
Original array: 1 3 5 7 9
Reversed array: 9 7 5 3 1
```

2. Write a program which performs the following tasks:
- Initialize an integer array of 10 elements in main()
 - Pass the entire array to a function modify()
 - In modify() multiply each element of array by 3
 - Return the control to main() and print the new array elements in main()

Sol:

```
#include <stdio.h>

// Function to modify the array elements by multiplying each element by 3
void modify(int arr[], int size) {
    for (int i = 0; i < size; ++i) {
        arr[i] *= 3;
    }
}

int main() {
    int arr[10];

    // Initializing the array in main
    printf("Enter 10 elements of the array:\n");
    for (int i = 0; i < 10; ++i) {
        scanf("%d", &arr[i]);
    }

    // Passing the array to the modify function
    modify(arr, 10);

    // Printing the modified array in main
    printf("Modified array elements:\n");
    for (int i = 0; i < 10; ++i) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

```
Enter 10 elements of the array:
1 2 3 4 5 6 7 8 9 11
Modified array elements:
3 6 9 12 15 18 21 24 27 33
```

Date: _____

Signature of faculty in-charge