ACPCE
Where knowledge is second nature

Jawahar Education Society's
A. C. Patil College of Engineering, Kharghar
Navi Mumbai 410210

Student Name: Manish Gangole

Course Name: C.S.E. (IoT CS BC)

Year: S.E.

Roll No.:

PRN No.: 221111028

Course code: CSL301

Semester: 3

Experiment Evaluation Sheet

Experiment No.: 1

Experiment Name:
Program to store the elements in 1-D array and perform the operations like searching, sorting, reversing the elements.

Sr No.	Evaluation Criteria	Marks (Out of 9)	Performance Date	Correction Date and Signature of Instructor
1	Experiment Performance			
2	Journal Performance			
3	Punctuality			
Total				

Code :

```
#include <stdio.h>

void storeElements(int arr[], int n) {
    printf("Enter the elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
}

int searchElement(int arr[], int n, int target) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == target) {
            return i;
        }
    }
    return -1;
}

void sortArray(int arr[], int n) {
    int temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("Array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void reverseArray(int arr[], int n) {
    int temp, start = 0, end = n - 1;
    while (start < end) {
        temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        start++;
        end--;
    }
    printf("Array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

Code :

```
int main() {
    int n, choice, target, index;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int arr[n];

    storeElements(arr, n);

    while (1) {
        printf("\nOperations:\n");
        printf("1. Search an element\n");
        printf("2. Sort the array\n");
        printf("3. Reverse the array\n");
        printf("4. Exit\n");

        printf("Enter your choice (1/2/3/4): ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the element to search: ");
                scanf("%d", &target);
                index = searchElement(arr, n, target);
                if (index != -1) {
                    printf("Element found at index %d\n", index + 1);
                } else {
                    printf("Element not found in the array.\n");
                }
                break;

            case 2:
                sortArray(arr, n);
                printf("Array sorted successfully.\n");
                break;

            case 3:
                reverseArray(arr, n);
                printf("Array reversed successfully.\n");
                break;

            case 4:
                printf("Exiting the program.\n");
                return 0;

            default:
                printf("Invalid choice. Please enter a valid option (1/2/3/4).\n");
        }
    }

    return 0;
}
```

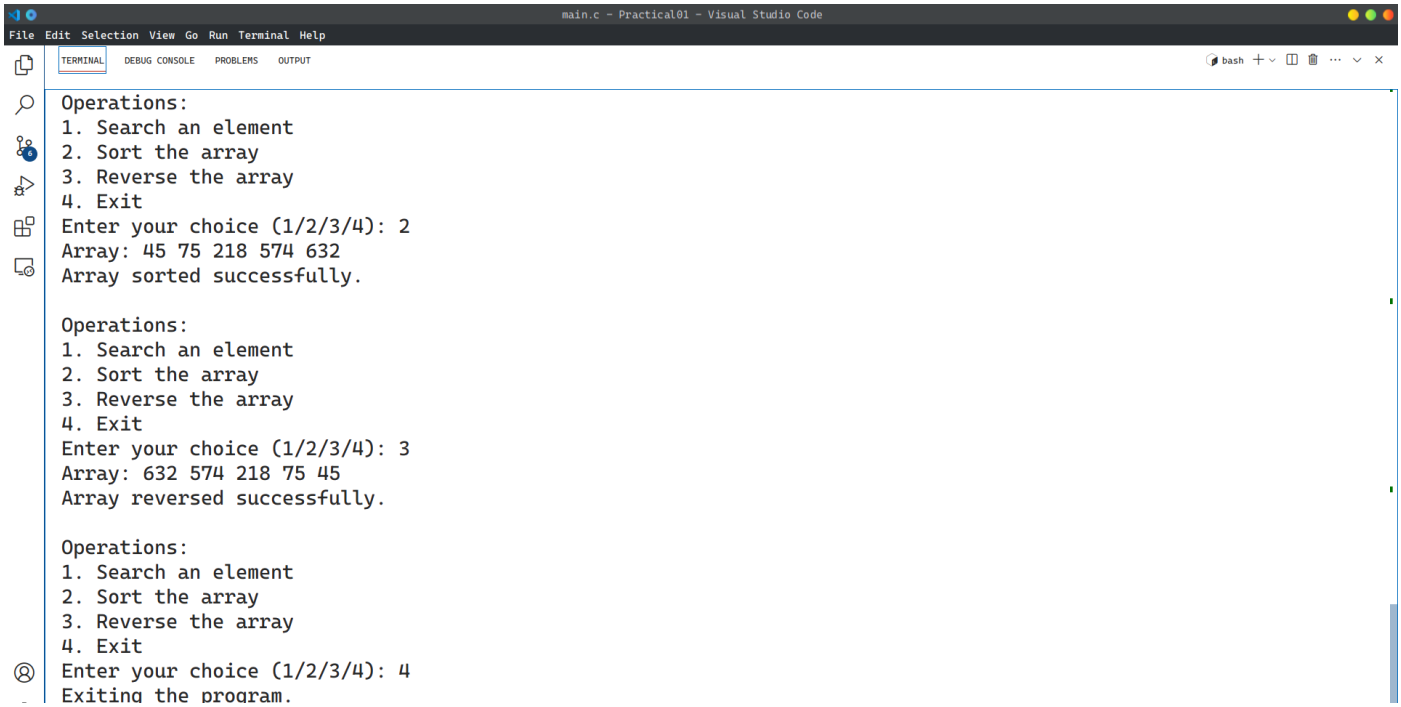
Output :

```
Enter the number of elements in the array: 5
Enter the elements:
45 632 75 218 574

Operations:
1. Search an element
2. Sort the array
3. Reverse the array
4. Exit
Enter your choice (1/2/3/4): 1
Enter the element to search: 45
Element found at index 1

Operations:
1. Search an element
2. Sort the array
3. Reverse the array
4. Exit
Enter your choice (1/2/3/4): 1
Enter the element to search: 997
Element not found in the array.

Operations:
1. Search an element
2. Sort the array
```



Conclusion :

Through this experiment we have learnt about how to implement an array using the C language.
Various operations like searching, sorting, and reversing are applied on the array.
This experiment helps us in using arrays as a data structure for further reference.