



East West University

Department of Computer Science & Engineering

A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212

Lab Manual : 01

Course Code : CSE207

Course Title : Data Structures

Instructor : Md. ManowarulIslam, Adjunct Faculty, Department of CSE

Objective:

The objective of this lab is to provide a fundamental idea about the sorting element of an integer array using C programming. At the end of the lab, students are able to know:

- How to take input into an array.
- How to sort the element of the array.

Preliminary:

Bubble sort is a sorting algorithm that compares two adjacent elements and swaps them until they are not in the intended order. We will learn how to take input from the user and find the output.

```
#include<stdio.h>
```

```
int main() {
```

```
    int a[100];
```

```
    int n, i, pos = -1, item;
```

```
    printf("Enter number of elements: ");
```

```
    scanf("%d", &n);
```

```
    printf("Enter your data :");
```

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

```
        scanf("%d", &a[i]);
```

```
    printf("Your array :");
```

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

```
        printf("%d\t", a[i]);
```

```
    int temp, j;
```

```
    for (i = 0; i < n - 1; i++)
```

```
        for (j = 0; j < n - 1 - i; j++)
```

```
        {
```

```
            if (a[j] > a[j + 1])
```

```
            {
```

```
                temp = a[j];
```

```

        a[j]=a[j+1];
        a[j+1]=temp;
    }
}

printf("\nYour array :");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
}

```

Lab Task

Exercise 1:

Modify the above code to count the number of pass and swap operation for an array of size N.

Exercise 2

Write a program to find second maximum and minimum number of an integer array. Show the sample input and output clearly.

Sample Input	Sample Output
17, 4,5,6,2,10	Second Maximum: 10 Second Minimum: 4

Exercise 3

The following algorithm will print the duplicate entry in an integer array.

Write a program in C to find the duplicate number.

ALGORITHM:

- **STEP 1:** START
- **STEP 2:** INITIALIZE arr[] = { 1, 2, 3, 4, 2, 7, 8, 8, 3 }.
- **STEP 3:** length = sizeof(arr)/sizeof(arr[0])
- **STEP 4:** PRINT "Duplicate elements in given array:"
- **STEP 5:** SET i=0. REPEAT STEP 6 to STEP 9 UNTIL i<length
- **STEP 6:** SET j=i+1. REPEAT STEP 7 and STEP 8 UNTIL i<length
- **STEP 7:** if(arr[i] == arr[j])
 PRINT arr[j]
- **STEP 8:** j=j+1
- **STEP 9:** i+1.
- **STEP 10:** RETURN 0.
- **STEP 11:** END

Sample Input	Sample Output
1, 2, 3, 4, 2, 7, 8, 8, 3	Duplicate elements in given array: 238

4. Write a program in C to delete an element at desired position from an array.

Sample Input	Sample Output
Input array elements: 10 20 30 40 50 Input position to delete: 2	Array elements: 10, 30, 40, 50

5. Write a program in C to insert element in array at specified position.

Sample Input	Sample Output
Input array elements: 10, 20, 30, 40, 50 Input element to insert: 25 Input position where to insert: 3	Elements of array are: 10, 20, 25, 30, 40, 50

6. Write a C program to check if an array is a palindrome or not. (Palindrome array is the one which reads the same forward and backward)

Sample Input / Output:

Enter the number of elements: 5

Enter the elements:

1 2 3 2 1

The array is a palindrome.

7. Write a C program to reverse the elements of an integer array.

Sample Input/Output:

Enter the number of elements: 5

Enter the elements:

1 2 3 4 5

The reversed array is: 5 4 3 2 1