

Green University of Bangladesh Department of Computer Science and Engineering

(CSE)

**Faculty of Sciences and Engineering Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)**

**Lab Report NO #03 Course Title: Data Structure Lab**

**Course Code: CSE 206 Section: D8**

**Lab Experiment Name:** Implementation of Bubble sort, Insertion sort, Selection sort.

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Rukonuzzaman Topu | 232002280 |

**Lab Date : 18/09/24**

**Submission Date : 24/09/24 Course Teacher’s Name : Md. Parvez Hossain**

**Lab Report Status Marks: …………………………………**

**.....................**

**Signature:**

**Comments: .............................................. Date: ..............................**

# INTRODUCTION

The purpose of this lab reports is to know the concepts of sorting of array in the C program. Here, we will see how we can sort an array. We are going to use three type of array sorting. In this lab report our aim is to solve some real-world problems efficiently.

# OBJECTIVES

The primary objectives of this lab report are as follows:

* + To be familiar with different type of sorting algorithms.
  + To learn problem solving techniques using C.

# IMPLEMENTATION

Task 1: Implement Bubble Sort algorithm using Arrays Solution:

#include <stdio.h>

int main() {

    int ar[100];

    int n, temp;

    printf("Enter the size of array: ");

    scanf("%d", &n);

    printf("Enter the elements of the array:\n");

    for(int i = 0; i < n; i++) {

        scanf("%d", &ar[i]);

    }

    for(int i = 0; i < n - 1; i++) {

        int swapped = 0;

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

            if(ar[j] > ar[j + 1]) {

                temp = ar[j];

                ar[j] = ar[j + 1];

                ar[j + 1] = temp;

                swapped = 1;

            }

        }

        if (!swapped) {

            break;

        }

    }

    printf("Sorted array: \n");

    for(int i = 0; i < n; i++) {

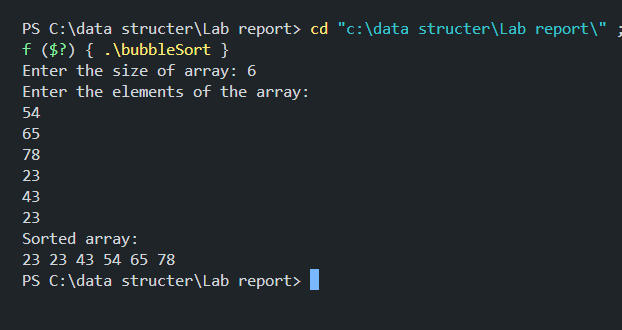
        printf("%d ", ar[i]);

    }

    printf("\n");

    return 0;

}



Task 2: Implement Insertion Sort algorithm using Arrays Solution:

#include <stdio.h>

int main() {

int ar[100]; int n,temp;

printf("Enter the size of aray:"); scanf("%d",&n);

for(int i=0; i<n; i++){ scanf("%d",&ar[i]);

}

for (int i = 1; i < n; i++) { int v = ar[i];

int j = i - 1;

while (j >= 0 && ar[j] > v) { ar[j + 1] = ar[j];

j--;

}

ar[j + 1] = v;

}

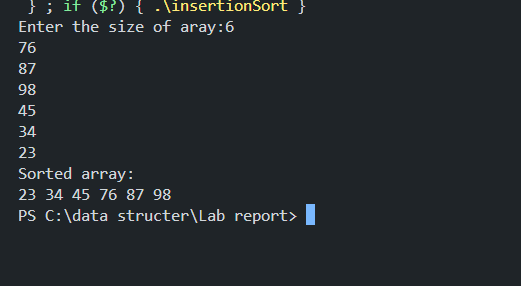
printf("Sorted array: \n"); for (int i = 0; i < n; i++) {

printf("%d ", ar[i]);

}

return 0;

}



Task 3: Implement Selection Sort algorithm using arrays. Solution:

#include <stdio.h>

int main() {

    int ar[100], n, temp;

    printf("Enter the size of the array: ");

    scanf("%d", &n);

    printf("Enter %d elements:\n", n);

    for (int i = 0; i < n; i++) {

        scanf("%d", &ar[i]);

    }

    for (int i = 0; i < n - 1; i++) {

        int min\_idx = i;

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

            if (ar[j] < ar[min\_idx]) {

                min\_idx = j;

            }

        }

        if (min\_idx != i) {

            temp = ar[min\_idx];

            ar[min\_idx] = ar[i];

            ar[i] = temp;

        }

    }

    printf("Sorted array: \n");

    for (int i = 0; i < n; i++) {

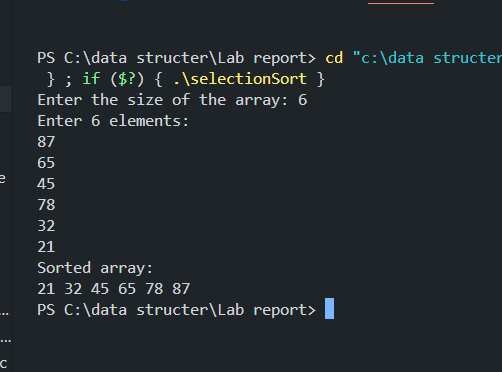
        printf("%d ", ar[i]);

    }

    printf("\n");

    return 0;

}



# DISCUSSION

In this lab report, we explored methods for sorting an array and tackled three different sorting algorithms. The first was the bubble sort, where we compared adjacent elements and swapped their positions if they met a certain condition. This process continues for a total of n2n^2n2 iterations. The second algorithm was insertion sort, which involves taking one element at a time from the unsorted portion of the array and placing it in the correct position within the sorted portion until the entire array is ordered. Lastly, we implemented selection sort, which repeatedly scans the array to identify the smallest remaining element and moves it to the front, continuing this process until the array is fully sorted.