Lab Report 7

Fall 2024

Course Title: Structured Programming Lab

Course Code: CSE 1202 (Fall 2024)

Submitted by:

Student Name and ID

Md. Tazminur Rahman Tanim (242014124)

Department of CSE

University of Liberal Arts Bangladesh (ULAB)

1. Write a program to find the maximum, minimum and average from a list of floating point numbers.

Answer :

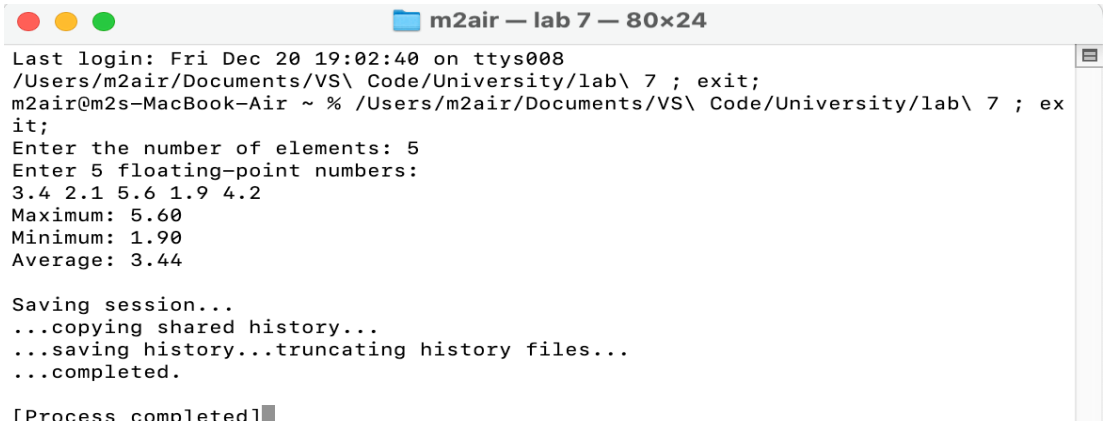
Algorithm:

- Input the size of the array.
- Input the floating-point numbers into the array.
- Initialize `max`, `min`, and a sum variable to compute the total.
- Traverse the array:
    - Update `max` if the current element is greater.
    - Update `min` if the current element is smaller.
    - Add the element to the sum.
- Compute the average as `sum / size`.
- Print the maximum, minimum, and average

Code

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    float numbers[n], sum = 0, max, min;
    printf("Enter %d floating-point numbers:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%f", &numbers[i]);
        sum += numbers[i];
        if (i == 0) {
            max = min = numbers[i];
        } else {
            if (numbers[i] > max) max = numbers[i];
            if (numbers[i] < min) min = numbers[i];
        }
    }

    printf("Maximum: %.2f\n", max);
    printf("Minimum: %.2f\n", min);
    printf("Average: %.2f\n", sum / n);

    return 0;
}
```

Output Result :



2. Write a program to split even and odd elements of an array into two arrays.

Answer :

Algorithm:

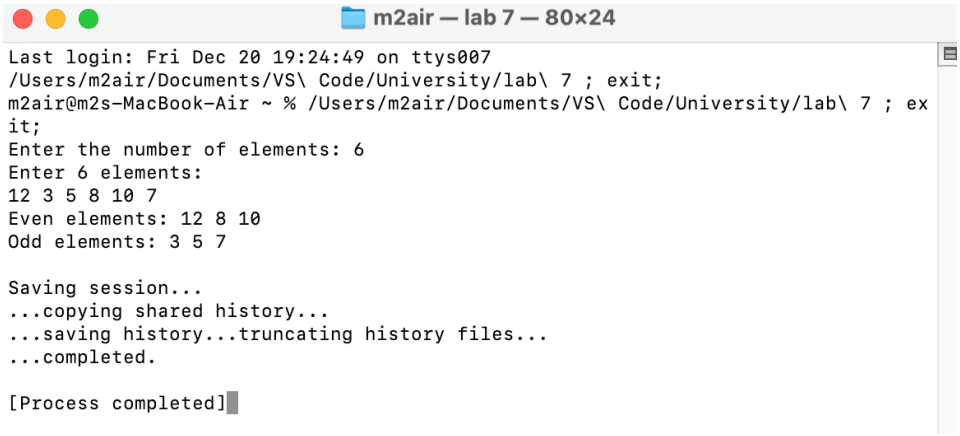
- Input the size of the array.
- Input the elements into the array.
- Traverse the array:
    - If the element is even, add it to the even array.
    - Otherwise, add it to the odd array.
- Print the even and odd arrays.

Code

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n], even[n], odd[n], e = 0, o = 0;
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        if (arr[i] % 2 == 0) {
            even[e++] = arr[i];
        } else {
            odd[o++] = arr[i];
        }
    }
```

```
    printf("Even elements: ");
    for (int i = 0; i < e; i++) printf("%d ", even[i]);
    printf("\n");

    printf("Odd elements: ");
    for (int i = 0; i < o; i++) printf("%d ", odd[i]);
    printf("\n");

    return 0;
}
```

Output Result :



3. Write a program to delete duplicate elements from an array

Answer :
Algorithm:

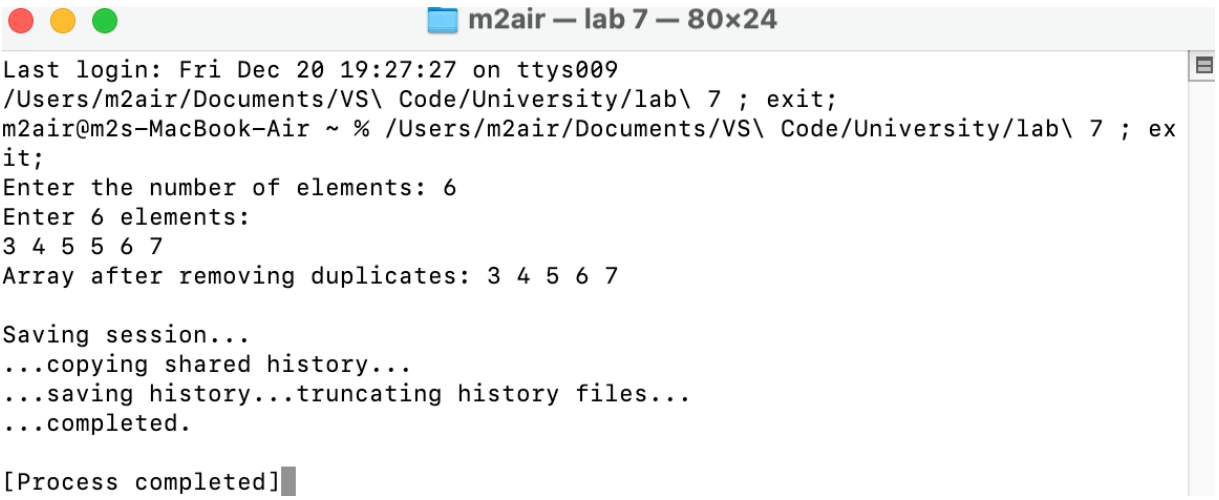
1. Input the size of the array.
2. Input the elements into the array.
3. Use nested loops:
    - Compare each element with the others.
    - If a duplicate is found, shift all subsequent elements to the left.
    - Reduce the size of the array.
4. Print the updated array.

Code

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of elements: ");
    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; i++) {
        for (int j = i + 1; j < n; ) {
            if (arr[i] == arr[j]) {
                for (int k = j; k < n - 1; k++) {
                    arr[k] = arr[k + 1];
                }
                n--;
            } else {
                j++;
            }
        }
    }

    printf("Array after removing duplicates: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

Output Result :



Discussion : The lab tasks focused on fundamental array operations in C programming. The first task computed the maximum, minimum, and average of floating-point numbers, emphasizing array traversal and mathematical operations. The second task separated even and odd elements into separate arrays, highlighting conditional checks. The third task removed duplicates, showcasing nested loops and in-place array manipulation. These exercises reinforced essential skills for solving real-world problems efficiently.