Assignment 15

Arrays and functions

```c
#include <stdio.h>

// Function to find the greatest number in an array of any size
int findMax(int arr[], int size) {
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];}
    }
    return max;
}

// Function to find the smallest number in an array of any size
int findMin(int arr[], int size) {
    int min = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] < min) {
            min = arr[i];}
    }
    return min;
}

// Function to sort an array of any size (using Selection Sort)
void sortArray(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        int minIndex = i;
```

```
        for (int j = i + 1; j < size; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j; }
        }
        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }
}

// Function to rotate an array by n positions to the left
void rotateLeft(int arr[], int size, int n) {
    int temp;
    for (int i = 0; i < n; i++) {
        temp = arr[0];
        for (int j = 0; j < size - 1; j++) {
            arr[j] = arr[j + 1];
        }
        arr[size - 1] = temp;
    }
}

// Function to find the first occurrence of adjacent duplicate values in the
array
int findAdjacentDuplicate(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        if (arr[i] == arr[i + 1]) {
            return arr[i];}
    }
```

```
    return -1;
}
// Function to display an array in reverse order
void displayReverse(int arr[], int size) {
    for (int i = size - 1; i >= 0; i--) {
        printf("%d ", arr[i]);
    }
    printf("\n\n");
}

// Function to count the total number of duplicate elements in an array
int countDuplicates(int arr[], int size) {
    int count = 0;
    for (int i = 0; i < size; i++) {
        for (int j = i + 1; j < size; j++) {
            if (arr[i] == arr[j]) {
                count++;
                break;}
        }
    }
    return count;
}

// Function to print all unique elements in an array
void printUniqueElements(int arr[], int size) {
    printf("Unique elements in the array: ");
    for (int i = 0; i < size; i++) {
        int isUnique = 1;
        for (int j = 0; j < size; j++) {
```

```
            if (i != j && arr[i] == arr[j]) {
                isUnique = 0;
                break;}
        }
        if (isUnique) {
            printf("%d ", arr[i]);}
    }
}

// Function to merge two arrays of the same size sorted in descending order
void mergeArrays(int arr1[], int arr2[], int size, int merged[]) {
    for (int i = 0; i < size; i++) {
        merged[i] = arr1[i];
    }
    for (int i = 0; i < size; i++) {
        merged[i + size] = arr2[i];
    }
    sortArray(merged, 2 * size);
}

// Function to count the frequency of each element in an array
void countFrequency(int arr[], int size) {
    printf("Element  Frequency\n");
    for (int i = 0; i < size; i++) {
        int count = 1;
        for (int j = i + 1; j < size; j++) {
            if (arr[i] == arr[j]) {
                count++;}
        }
```

```
            printf("%d\t%d\n", arr[i], count);
            i += count - 1;
        }
    }
    // Driver
    int main() {
        int arr[100], size, n, d, arr2[100], merged[200];
        printf("Enter the size of the array (up to 100): ");
        scanf("%d", &size);
        printf("Enter %d elements of the array: ", size);
        for (int i = 0; i < size; i++) {
            scanf("%d", &arr[i]);
        }

        printf("The greatest number in the array is: %d\n\n", findMax(arr, size));

        printf("The smallest number in the array is: %d\n\n", findMin(arr, size));

        sortArray(arr, size);
        printf("Sorted array: ");
        for (int i = 0; i < size; i++) {
            printf("%d ", arr[i]);
        }
        printf("\n\n");

        printf("Enter the number of positions to rotate: ");
        scanf("%d", &d);
        printf("Enter the direction (left or right, 'L' or 'R'): ");
        char direction;
```

```
    scanf(" %c", &direction);
    if (direction == 'L' || direction == 'l') {
        rotateLeft(arr, size, d);}
    printf("Array after rotation: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n\n");

    int adjacentDuplicate = findAdjacentDuplicate(arr, size);
    if (adjacentDuplicate != -1) {
        printf("First adjacent duplicate: %d\n\n", adjacentDuplicate);
    } else {
        printf("No adjacent duplicates found in the array.\n\n");
    }

    printf("Array in reverse order: ");
    displayReverse(arr, size);

    printf("Total number of duplicate elements: %d\n\n", countDuplicates(arr,
size));
    printUniqueElements(arr, size);

    printf("Enter the second array of the same size: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr2[i]);
    }
    mergeArrays(arr, arr2, size, merged);
    printf("Merged and sorted array in descending order:\n ");
```

```c
    for (int i = 0; i < 2 * size; i++) {
        printf("%d ", merged[i]);
    }
    printf("\n\n");

    countFrequency(arr, size * 2);
    return 0;
}
```

```
Enter the size of the array (up to 100): 10
Enter 10 elements of the array: 88
17
53
53
67
88
1
49
73
19
The greatest number in the array is: 88

The smallest number in the array is: 1

Sorted array: 1 17 19 49 53 53 67 73 88 88

Enter the number of positions to rotate: 7
Enter the direction (left or right, 'L' or 'R'): L
Array after rotation: 73 88 88 1 17 19 49 53 53 67

First adjacent duplicate: 88

Array in reverse order: 67 53 53 49 19 17 1 88 88 73

Total number of duplicate elements: 2

Unique elements in the array: 73 1 17 19 49 67
```

```
Enter the second array of the same size: 15
19
19
4
20
69
42
58
87
62
Merged and sorted array in descending order:
 1 4 15 17 19 19 19 20 42 49 53 53 58 62 67 69 73 87 88 88

Element  Frequency
1        1
4        1
15       1
17       1
19       3
20       1
42       1
49       1
53       2
58       1
62       1
67       1
69       1
73       1
87       1
88       2
```