

Assessment 1

1. Write a C program to separate even and odd numbers from an array into two different arrays. Extend the problem with additional requirements to handle edge cases, dynamic inputs, and advanced validations, while utilizing conditional statements for classification.

Detailed Requirements:

Input Constraints:

Accept the size of the array NNN from the user. Validate NNN to be between 10 and 200, inclusive.

Populate the array with NNN integers entered by the user. Allow both positive and negative integers, including zeros.

Classification Rules:

Classify numbers as even or odd using conditional statements (if/else).

Place all even numbers in one array and all odd numbers in another.

Maintain the order of numbers as they appear in the input array.

Additional Features:

Count the total number of even and odd numbers separately.

Display the indices of even and odd numbers from the original array.

Compute the sum and average of even and odd numbers separately.

Edge Case Handling:

If all numbers are even, display "No odd numbers found in the array."

If all numbers are odd, display "No even numbers found in the array."

Handle arrays with repeated numbers correctly without additional checks for uniqueness.

Output Requirements:

Display the original array.

Display the arrays of even and odd numbers.

Show the indices of even and odd numbers in the original array.

Print the count, sum, and average of numbers in each category (even and odd).

Extended Functionality:

After processing the first array, allow the user to input another array without restarting the program.

If the user enters invalid input for the array size or elements, prompt them to re-enter until valid input is provided.

Example Execution:

Input 1:

Array Size (N): 10

Array Elements: [12, -3, 0, 45, 28, 19, -10, 34, 7, 2]

Processing:

Even Numbers: [12, 0, 28, -10, 34, 2]

Indices: [0, 2, 4, 6, 7, 9]

Count: 6

Sum: 66

Average: 11

Odd Numbers: [-3, 45, 19, 7]

Indices: [1, 3, 5, 8]

Count: 4

Sum: 68

Average: 17

Output:

Original Array: [12, -3, 0, 45, 28, 19, -10, 34, 7, 2]

Even Numbers: [12, 0, 28, -10, 34, 2]

Indices: [0, 2, 4, 6, 7, 9]

Count: 6

Sum: 66

Average: 11

Odd Numbers: [-3, 45, 19, 7]

Indices: [1, 3, 5, 8]

Count: 4

Sum: 68

Average: 17

Input 2:

Array Size (N): 6

Array Elements: [1, 3, 5, 7, 9, 11]

Output:

Original Array: [1, 3, 5, 7, 9, 11]

Even Numbers: None (No even numbers found in the array).

Odd Numbers: [1, 3, 5, 7, 9, 11]

Indices: [0, 1, 2, 3, 4, 5]

Count: 6

Sum: 36

Average: 6

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

```
#define MIN_SIZE 10
```

```
#define MAX_SIZE 200
```

```
// Function to calculate sum
```

```
int sum_cal(int arr[], int size) {
```

```
    int sum = 0;
```

```
    for (int i = 0; i < size; i++) {
```

```
        sum += arr[i];
```

```
    }
```

```
    return sum;
```

```
}
```

```

// Function to calculate average
float avg_cal(int sum, int size) {
    return (float)sum / size;
}

int main() {
    int size;

    int arr[MAX_SIZE], odd[MAX_SIZE], even[MAX_SIZE];
    int even_indices[MAX_SIZE], odd_indices[MAX_SIZE];
    int even_count = 0, odd_count = 0;

    // Input array size
    do {
        printf("Enter the size of the array (10-200): ");
        scanf("%d", &size);

        if (size < MIN_SIZE || size > MAX_SIZE) {
            printf("Invalid size. Please try again.\n");
        }
    } while (size < MIN_SIZE || size > MAX_SIZE);

    // Input array elements
    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        printf("Element [%d]: ", i);
        scanf("%d", &arr[i]);
    }

    // Classify elements into even and odd
    for (int i = 0; i < size; i++) {

```

```

    if (arr[i] % 2 == 0) {
        even[even_count] = arr[i];
        even_indices[even_count] = i;
        even_count++;
    } else {
        odd[odd_count] = arr[i];
        odd_indices[odd_count] = i;
        odd_count++;
    }
}

// Display original array
printf("\nOriginal Array: ");
for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);
}
printf("\n");

// Display even numbers
if (even_count > 0) {
    printf("\nEven Numbers: ");
    for (int i = 0; i < even_count; i++) {
        printf("%d ", even[i]);
    }
    printf("\nIndices: ");
    for (int i = 0; i < even_count; i++) {
        printf("%d ", even_indices[i]);
    }
    int even_sum = sum_cal(even, even_count);

```

```

        float even_avg = avg_cal(even_sum, even_count);

        printf("\nCount: %d\nSum: %d\nAverage: %.2f\n", even_count, even_sum,
even_avg);
    } else {
        printf("\nNo even numbers found in the array.\n");
    }

// Display odd numbers
if (odd_count > 0) {
    printf("\nOdd Numbers: ");
    for (int i = 0; i < odd_count; i++) {
        printf("%d ", odd[i]);
    }
    printf("\nIndices: ");
    for (int i = 0; i < odd_count; i++) {
        printf("%d ", odd_indices[i]);
    }
    int odd_sum = sum_cal(odd, odd_count);
    float odd_avg = avg_cal(odd_sum, odd_count);
    printf("\nCount: %d\nSum: %d\nAverage: %.2f\n", odd_count, odd_sum,
odd_avg);
} else {
    printf("\nNo odd numbers found in the array.\n");
}

return 0;
}

```