Abhiram's Assessment

Problem Statement: Advanced In-Place Array Reversal

Objective:
Write a C program to reverse the elements of a single-dimensional array in-place using a while loop. The program must account for the following additional requirements:

1. **Input Constraints**:
    - The array should support both integer and floating-point values, determined dynamically by the user.
    - The array size (N) should be between 5 and 100, inclusive, and must be validated. If the input is out of bounds, prompt the user until a valid size is entered.
2. **Reversal Conditions**:
    - For integer arrays, only even-indexed elements should be reversed.
    - For floating-point arrays, reverse the elements that have an absolute value greater than a user-defined threshold (T). Elements not meeting the threshold should remain in their original positions.
3. **Loop Constraints**:
    - Use a single while loop to implement the reversal logic. Nested loops are not allowed for the reversal operation.
    - The loop should terminate as soon as all eligible elements are reversed.
4. **Edge Case Handling**:
    - If no elements meet the reversal condition (e.g., all indices are odd for integers or all float values are below the threshold), the program should indicate that no reversal was performed.
    - For arrays with an odd number of elements, the middle element (if eligible) should remain untouched.
5. **Output Requirements**:
    - Display the original array and the reversed array after the operation.
    - Highlight the reversed elements by printing their indices and values separately.
6. **Example Execution**:
    - **Input**:
      Array Type: Integer
      Array: [3, 8, 7, 4, 9, 2]
      Output:
      Original Array: [3, 8, 7, 4, 9, 2]
      Reversed Array: [2, 8, 7, 4, 9, 3]
      Reversed Indices: 0 ↔ 5

Input:
Array Type: Floating-point

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

void reverse_int_array(int *arr, int size){
   int start = 0, end = size -1;
   int reversed = 0;
   while(start<end);{
      while(start < size && start % 2 != 0);
   start++;
      while (end>=0 && end % 2 !=0)
   end--;
      if(start<end){
         int temp = arr[start];
         arr[start]=arr[end];
         arr[end]=temp;
         start++;
         end--;
         reversed = 1;
      }
   }
if(!reversed){
   printf("\n No elements meets the condition of the threshold\n");
}
}
void reverse_float_array(float *arr, int size, float threshold){
   int start=0, end = size -1;
   int reversed = 0;
```

```
    while(start<end){
        while(start<size && fabs(arr[start])<= threshold) start++;
        while(end>=0 && fabs(arr[end])<=threshold) end--;
        if (start<end){
            float temp = arr[start];
            arr[start]=arr[end];
            arr[end]=temp;
            start++;
            end--;
            reversed=1;
        }
    }
    if (reversed){
        printf("\n no elements met the reverse condition)\n");
    }
}
int main(){
    int n,i,ch;
    printf("Enter the size of the array (5-100):");
    while(scanf("%d", &n), n < 5 || n > 100);
{
printf("invalid size, enter size bw 5 & 100:");
}
printf("choose array type (interger 1, floating pt 2:");
scanf("%d", &ch);
if (ch==1){
    int arr[n];
    printf("enter %d integers:",n);
    for (i=0; i<n; i++){
        scanf("%d", &arr[i]);
    }
```

```
    printf("Original Array:");
    for (i=0;i < n; i++){
        printf("%d", arr[i]);
    }
    reverse_int_array(arr,n);
    printf("\nreversed array:");
    for(i=0;i<n;i++){
        printf("%d", arr[i]);
    }
    printf("\nReversed indices:");
    for (i=0; i < n; i++){
        if(i % 2 ==0) printf("%d" ,i);
    }
}
else if (ch==2){
    float arr[n], threshold;
    printf("enter %d floating pt numbers" ,n);
    for (i=0; i < n; i++){
        scanf("%f", &arr[i]);
    }
    printf("Enter threshold value:");
    scanf("%f", &threshold);
    printf("Original Array:");
    for (i=0; i<n; i++){
        printf("%2f", arr[i]);
    }

    reverse_float_array(arr, n, threshold);
    printf("\nReversed array:");
    for (i = 0; i<n; i++){
        printf("%2f", arr[i]);
```

```
    }
    printf("\nReversed Indices:");
    for (i=0; i<n; i++){
        if (fabs(arr[i]> threshold));
    printf("%d", i);
    }
}
else {
    printf("invalid choice is entered\n");
}
return 0;
}
```