---Assesment 1---

Q. *Write a C program to identify all perfect numbers in an array of integers. A perfect number is a positive integer that is equal to the sum of its proper divisors (excluding itself). For example, 6 is a perfect number because its divisors are 1, 2, and 3, and their sum is 6. Similarly, 28 is also a perfect number because its divisors are 1, 2, 4, 7, and 14, and their sum is 28.*

Detailed Requirements:

Input Constraints:

The program should take an integer array as input, with the array size (N) specified by the user.

The array size should be between 5 and 100, inclusive. Validate the input and prompt the user if it's invalid.

All array elements must be integers, including negative numbers. Only positive numbers should be checked for perfection.

Identification Logic:

Use a loop to iterate through the array.

For each positive number in the array, use a nested loop to find its divisors.

Use an if condition to verify whether the sum of its divisors equals the number itself.

Performance Optimization:

Optimize the divisor sum calculation by iterating only up to the square root of the number. If i is a divisor, include both i and number/i in the sum.

Output Requirements:

Display all perfect numbers found in the array along with their positions (indices) in the array.

For each perfect number, list its divisors explicitly.

If no perfect numbers are found, display an appropriate message.

Edge Case Handling:

Ignore negative numbers and zero in the array during the computation.

If the array contains duplicates, count and display each occurrence of the perfect number separately.

Example Execution:

Input:

Array: [6, -3, 28, 12, 496, 10]

Output:

Perfect Number: 6 at Index 0

Divisors: [1, 2, 3]

Perfect Number: 28 at Index 2

Divisors: [1, 2, 4, 7, 14]

Perfect Number: 496 at Index 4

Divisors: [1, 2, 4, 8, 16, 31, 62, 124, 248]

No Perfect Numbers Message: If the array contains no perfect numbers, display:

"No perfect numbers found in the given array."

Key Example:

Perfect Number:

6: Divisors are 1, 2, and 3. Their sum is 1+2+3=61 + 2 + 3 = 61+2+3=6.

28: Divisors are 1, 2, 4, 7, and 14. Their sum is 1+2+4+7+14=281 + 2 + 4 + 7 + 14 = 281+2+4+7+14=28.

496: Divisors are 1, 2, 4, 8, 16, 31, 62, 124, 248. Their sum is 1+2+4+8+16+31+62+124+248=4961 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 = 4961+2+4+8+16+31+62+124+248=496.

---CODE--

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

bool perfect_number(int n) {
    if (n <= 0) {
        return false;
    }
    int sum = 0;
    int divisors[50];
    int divisor_count = 0;
    for (int i = 1; i <= sqrt(n); i++) {
        if (n % i == 0) {
            sum += i;
            divisors[divisor_count++] = i;
            if (i != 1 && i != n / i) {
                sum += n / i;
                divisors[divisor_count++] = n / i;
            }
        }
    }

    if (sum == n) {
        printf("The divisors are: ");
        for (int i = 0; i < divisor_count; i++) {
            printf("%d", divisors[i]);
            if (i < divisor_count - 1) {
                printf(", ");
            }
        }
        printf("\n");
        return true;
    }
    return false;
}
```

```
int main() {
    int size;

    do {
        printf("Enter the number of elements (between 5 and 100): ");
        scanf("%d", &size);
        if (size < 5 || size > 100) {
            printf("Invalid input! Please enter a number between 5 and 100.\n");
        }
    } while (size < 5 || size > 100);

    int *arr = (int *)malloc(size * sizeof(int));

    for (int i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    bool found_perfect = false;

    for (int i = 0; i < size; i++) {
        if (arr[i] > 0 && perfect_number(arr[i])) {
            printf("Perfect Number: %d at Index %d\n", arr[i], i);
            found_perfect = true;
        }
    }
    if (!found_perfect) {
        printf("No perfect numbers found in the given array.\n");
    }

    free(arr);
    return 0;
}
```

```
--------------------------------------------------------OUTPUT--------------------------------------
Enter the number of elements (between 5 and 100): 6
Enter element 1: 6
Enter element 2: 5
Enter element 3: 4
Enter element 4: 3
Enter element 5: 28
Enter element 6: 30
The divisors are: 1, 2, 3
Perfect Number: 6 at Index 0
The divisors are: 1, 2, 14, 4, 7
Perfect Number: 28 at Index 4
```