STUDENT REPORT

DETAILS

Name

Ganesh B

Roll Number

3BR23EE028

EXPERIMENT

Title

NUMBER OF COMBINATIONS LEADING TO A PRODUCT

Description

Problem Statement:

You are given an array arr and a product m. Your task is to find the number of possible unique triplets whose product of elements is m.

Input Format:

- The first line contains the integer, n
- The second line contains space seperated integers of the array, arr
- The third line contains the product m.

The input will be read from the STDIN by the candidate

Output Format:

The output consists of a single integer, i.e. the count of unique triplets having product m.

The output will be matched to the candidate's output printed on the STDOUT

Example:

Input:

7

5 3 20 10 1 4 2

60

Output:

3

Explanation:

Product m:60

Possible triplets for product m: (5,4,3),(20,3,1), (10,3,2)

The count of unique triplets is 3.

Source Code:

```
def count_unique_triplets(arr, m):
    arr.sort()
    unique_triplets = set()
    n = len(arr)
    for i in range(n):
        if i > 0 and arr[i] == arr[i - 1]:
            continue
        target = m / arr[i]
        left, right = i + 1, n - 1
        while left < right:
            product = arr[left] * arr[right]
            if product < target:
                left += 1
            elif product > target:
                right -= 1
            else:
                unique_triplet = (arr[i], arr[left], arr[right])
                unique_triplets.add(unique_triplet)
                left += 1
                right -= 1
                while left < right and arr[left] == arr[left - 1]:
                    left += 1
                while left < right and arr[right] == arr[right + 1]:
                    right -= 1

    return len(unique_triplets)
if __name__ == "__main__":
    import sys

    n = int(sys.stdin.readline().strip())
    arr = list(map(int, sys.stdin.readline().strip().split()))
    m = int(sys.stdin.readline().strip())

    result = count_unique_triplets(arr, m)
    print(result)
```

RESULT

6 / 6 Test Cases Passed | 100 %