Logo

STUDENT REPORT

DETAILS

Name

SEEMA

Roll Number

3BR23CS140

EXPERIMENT

Title

SUM OF NUMBERS AT PRIME FACTORS

Description

Prime factors of a positive integer are the prime numbers that divide that integer exactly.

Given an array arr of n integers and a positive integer num.

Let's suppose prime factorization of num is: p^a x q^b x r^c x .... x z^f ,where p,q,r...z are prime numbers.

Sum of numbers in array arr at indices of prime factors of number num is: a x arr[p] + b x arr[q] + c x arr[r] +...... + f x arr[z].

You are given an array arr of size n and a positive integer num. You are required to calculate the sum of numbers in arr as mentioned above, and print the same.

Note:

- If arr is empty, print -1.
- If prime factor of num not found as indices, print 0.

Input Format:

The input consists of three lines:

- The first line contains an integer, i.e. n.
- The second line contains an array arr of length of n.
- The third line contains an integer num

The input will be read from the STDIN by the candidates.

Output Format:

Print the sum that was mentioned in the problem statement.

Example:

Input:

6

11 21 32 45 1 23

6

Output:

77

Explanation:

$6=2^1$ x 3^1

sum=1*arr[2]+1*arr[3]=1*32+1*45=77

Source Code:

```
def prime_factorization_sum(arr, num):
  """Calculates the sum of numbers in arr at indices of prime factors of num.

  Args:
    arr: A list of integers.
    num: A positive integer.

  Returns:
    The sum of numbers in arr at indices of prime factors of num.
  """

  prime_factors = []
  while num > 1:
    for i in range(2, int(num**0.5) + 1):
      if num % i == 0:
        prime_factors.append(i)
        num //= i
        break
    else:
      prime_factors.append(num)
      break

  return sum(arr[i] for i in prime_factors if i < len(arr))

# Example usage:
n=int(input())
arr = list(map(int,input().split()))
num = int(input())
result = prime_factorization_sum(arr, num)
print(result)  # Output: 9 (arr[2]
```

RESULT

3 / 5 Test Cases Passed | 60 %