Logo

STUDENT REPORT

DETAILS

Name

BANDU BHAI MOHAMMED EISSA

Roll Number

TEMPBTech-ECE002

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: $p^a \times q^b \times r^c \times \dots \times z^f$,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:

$6 = 2^1 \times 3^1$

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

Source Code:

```python
def prime_factors(n):
    factors = {}
    # Check for number of 2s that divide n
    while n % 2 == 0:
        if 2 in factors:
            factors[2] += 1
        else:
            factors[2] = 1
        n //= 2

    # Check for odd factors from 3 onwards
    for i in range(3, int(n**0.5) + 1, 2):
        while n % i == 0:
            if i in factors:
                factors[i] += 1
            else:
                factors[i] = 1
            n //= i

    # If n is a prime number greater than 2
    if n > 2:
        factors[n] = 1

    return factors

def calculate_weighted_sum(n, arr, num):
    if n == 0:
        return -1

    # Get the prime factorization of num
    factors = prime_factors(num)

    total_sum = 0
    found_valid_index = False

    # Calculate the sum based on prime factor indices
    for prime, count in factors.items():
        if prime < n:   # Check if prime can be used as an index
            total_sum += count * arr[prime]
            found_valid_index = True

        if prime>=n:
            return 0

    return total_sum

# Read input
n = int(input().strip())
if n == 0:
    print(-1)
else:
    arr = list(map(int, input().strip().split()))
    num = int(input().strip())
    result = calculate_weighted_sum(n, arr, num)
    print(result)
```

RESULT

5 / 5 Test Cases Passed | 100 %