



STUDENT REPORT

DETAILS

Name

THABITHA K

Roll Number

TEMPBTech-CSE083

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:

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

$$\text{sum}=1*\text{arr}[2]+1*\text{arr}[3]=1*32+1*45=77$$

Source Code:

```
def prime_factors(n):
    factors = set()
    # Check for number of 2s that divide n
    while n % 2 == 0:
        factors.add(2)
        n //= 2

    # n must be odd at this point so a skip of 2 (i.e., i = i + 2) can be used
    for i in range(3, int(n**0.5) + 1, 2):
        while n % i == 0:
            factors.add(i)
            n //= i

    # This condition is to check if n is a prime number greater than 2
    if n > 2:
        factors.add(n)

    return list(factors)

def calculate_sum(arr, num):
    if not arr: # Check if the array is empty
        return -1

    factors = prime_factors(num) # Get prime factors of num
    total_sum = 0
    found_valid_index = False

    for factor in factors:
        if factor < len(arr): # Check if factor can be used as an index
            total_sum += arr[factor]
            found_valid_index = True

    return total_sum if found_valid_index else 0 # Return sum or 0 if no valid index found

# Reading input
n = int(input()) # Length of the array
arr = list(map(int, input().split())) # The array elements
num = int(input()) # The number to factor

# Calculate and print the result
result = calculate_sum(arr, num)
print(result)
```

RESULT

2 / 5 Test Cases Passed | 40 %