



# STUDENT REPORT

## DETAILS

### Name

MAMATA PITTER

### Roll Number

KUB23CSE075

## 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:  $a \times arr[p] + b \times arr[q] + c \times arr[r] + \dots + f \times arr[z]$ .

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:

```
from collections import defaultdict
def prime_factors(num):
    factors=defaultdict(int)
    while num%2==0:
        factors[2]+=1
        num//=2
    for i in range(3,int(num**0.5)+1,2):
        while num%i == 0:
            fsctors[i]+=1
            num//=i
    if num>2:
        factors[num]+=1
    return factors
def calculate_prime_index_sum(arr,num):
    if not arr:
        return-1
    factors=prime_factors(num)
    total_sum=0
    valid_prime_found=False
    for prime,power in factors.items():
        if prime
```

### RESULT

0 / 5 Test Cases Passed | 0 %