Logo

STUDENT REPORT

DETAILS

Name

MOHAMMED FURKHAN

Roll Number

KUB23CSE085

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

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

    # Check for odd factors from 3 to sqrt(n)
    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(arr, num):
    if not arr:
        return -1

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

    # Initialize the sum
    weighted_sum = 0

    # Calculate the weighted sum based on prime factors and their counts
    for prime, count in factors.items():
        index = prime - 1  # Convert to zero-based index
        if 0 <= index < len(arr):  # Check if index is valid
            weighted_sum += count * arr[index]

    return weighted_sum

# Input processing
if __name__ == "__main__":
    n = int(input().strip())
    arr = list(map(int, input().strip().split()))
    num = int(input().strip())

    result = calculate_weighted_sum(arr, num)
    print(result)
Explanation:
Function prime_factors(n):

This function calculates the prime factors of n and their counts. It returns a dictionary where keys are the prime f
actors and values are their corresponding counts.
Function calculate_weighted_sum(arr, num):

This function
```

RESULT

0 / 5 Test Cases Passed | 0 %