6-N-Cryption

In a software company a security team is working on an encryption system called 6-N-Cryption. Their task is to encrypt a data by using dynamic number of keys. Then encrypt those keys again by a secret key. Then they store them to a Database.
You can assume that the original data is always an integer.

Encrypt Original Data:
The original data is encrypted by adding some keys with it. All the keys are prime numbers and none of the keys are used more than once. Also it is guaranteed that in every encryption they must use at least 2 keys.

Encrypt the key:
Every key is encrypted by multiplying it with a big prime number, **p**. This big prime number must be greater than all the keys.

Then the encrypted data and keys are stored in the database.
Now, you are sitting in their office for an interview and they give you this problem. You have to find the original data.

Sample Encrypt process with 2 keys:
Original data + Key_1 + Key_2 = Encrypted data (E).
Then the team generates a big prime number 'p' that has to be greater than all keys to encrypt all of them.
Key_1 * p = Encrypted_Key_1
Key_2 * p = Encrypted_Key_2

Sample Decrypt process with 2 keys:
Determine the big prime number, **p**. Then divide those Encrypted_Keys to get the real keys.
Key_1 = Encrypted_Key_1 / p;
Key_2 = Encrypted_Key_2 / p;
Original data = Encrypted data - Key_1 - Key_2;

Input:

Each test case start with an encrypted number ,**E**. Then the number of keys N(1<**N**<=10). Then there are N lines of input for the encrypted keys **EK**(1< **EK** < 2^63)
Two successive cases are separated by a single new line.

Output:

For each test case, print "Output : O" (without quotation) where **O** denotes the original data.

Sample Input:

29
2
14
21

61
3
481
703
185

58
5
253
299
391
115
46

119
4
7199
939
13459
3443

Sample Output:

Output: 24
Output: 24
Output: 10
Output: 39

Explanation of First Test Case:

Encrypted data is 29.

*In this case we have found **7** as big prime number. Then we divide those encrypted keys by this big prime number.*

14 / 7 = 2

21 / 7 = 3

Original data = 29 - 2 - 3 = 24.