

QUESTION1

The original recipe contains only whole numbers of ingredients, determine how much of each ingredient to use in order to make as little food as possible, which has the same ratios of ingredients, reduced recipe to only contain whole numbers of ingredients as well.

Input Format:

The first line of the input consists of a positive integer N, which corresponds to the number of ingredients.

Next line contains N space-separated integers, each indicating the quantity of a particular ingredient that is used.

Output Format:

Output exactly N space-separated integers on a line that gives the quantity of each ingredient that should use in order to make as little food as possible.

Sample Input 1:

```
2
4 4
```

Sample Output 1:

```
1 1
```

Sample Input 2:

```
3
2 3 4
```

Sample Output 2:

```
2 3 4
```

QUESTION2

A number is an AES number if it has exactly four divisors. For example, 10 is an AES number because it has exactly four divisors (1, 2, 5, 10). 12 is not an AES number because it has too many divisors (1, 2, 3, 4, 6, 12).

Given a range of numbers, write a program that counts how many numbers from that range are AES numbers.

Input Format:

The input consists of 2 space-separated integers, which corresponds to the

lower limit and the upper limit of the number range.

You may assume that the numbers in the range are less than 1000.

Output Format:

Output a single line that gives the count of AES numbers from the given range.

Sample Input 1:

1 20

Sample Output 1:

5

Sample Input 2:

50 100

Sample Output 2:

17

QUESTION3

A first year student will buy a PINK ticket, a second year student will buy a GREEN ticket, a third year student will buy a RED ticket, and a fourth year student will buy an ORANGE ticket.

Assume that all tickets are sold. Each colour of ticket is uniquely priced. Write a program to output all combinations of tickets that produce exactly the desired amount to be raised. Also display the total number of combinations found and the smallest number of tickets to be printed to raise the desired amount so that printing cost is minimized.

Input Format:

First 4 lines of the input correspond to the cost of a PINK, GREEN, RED, ORANGE ticket (in the exact order).

Last line of the input corresponds to the exact amount of money to be raised by selling tickets.

Output Format:

Output all combinations of tickets that produce exactly the desired amount to be raised. The combinations may appear in any order. Output the total number of combinations found. Output the smallest number of tickets to print to raise the desired amount so that printing cost is minimized.

Refer sample input and output for formatting specifications.

Sample Input 1:

1
2
3
4
3

of PINK is 0 # of GREEN is 0 # of RED is 1 # of ORANGE is 0
of PINK is 1 # of GREEN is 1 # of RED is 0 # of ORANGE is 0
of PINK is 3 # of GREEN is 0 # of RED is 0 # of ORANGE is 0
Total combinations is 3
Minimum number of tickets to print is 1

Sample Input 2:

1
2
4
3
4

Sample Output 2:

of PINK is 0 # of GREEN is 0 # of RED is 1 # of ORANGE is 0
of PINK is 0 # of GREEN is 2 # of RED is 0 # of ORANGE is 0
of PINK is 1 # of GREEN is 0 # of RED is 0 # of ORANGE is 1
of PINK is 2 # of GREEN is 1 # of RED is 0 # of ORANGE is 0
of PINK is 4 # of GREEN is 0 # of RED is 0 # of ORANGE is 0
Total combinations is 5
Minimum number of tickets to print is 1

QUESTION4

The number of events that the company organizes every month is recorded sensibly and is seemed to have followed a specific series like: 30, 35, 38, 41, 54, 53 ...

Write a program which takes an integer N as the input and will output the series till the Nth term.

Input Format:

First line of the input is an integer N.

Output Format:

Output a single line the series till Nth term, each separated by a comma.

Sample Input 1:

5

Sample Output 1:

30 35 38 41 54

Sample Input 2:

10

Sample Output 2:

30 35 38 41 54 53 78 71 110 95

QUESTION5

The academy has **N** different problem sets and there are N students who are appearing for the test. Each student is given exactly one problem set. But the problemsets might contain different number of problems in each one, which looks unfair. Move some problems from one problem set to another problemset (can do this multiple times), until all problem sets contain the same number of problems. Sometimes it's impossible to do so, in that case delete some problems completely before moving any problem, then start moving the problems.

Write a program to calculate the minimum number of problems to be deleted, and the minimum number of problems to be moved

Input Format:

First line of the input contains one integer N ($1 \leq N \leq 100$) representing the number of problemsets and the number of trainees.

Second line of the input contains N non-negative integers separated by a single space, representing the number of problems in each problemset. Each problemset will contain at most 100 problems.

Output Format:

Output a single line which will contain the minimum number of problems to be deleted, followed by a space then the minimum number of problems to be moved.

Sample Input 1:

3

3 3 3

Sample Output 1:

0 0

Sample Input 2:

4

6 4 5 3

Sample Output 2:

2 1