

Infytq\_coding\_questions

# Question 1: Biggest Meatball

- **Problem Statement** – Bhojon is a restaurant company and has started a new wing in a city. They have every type of cook except the meatball artist. They had fired their last cook because the sale of meatballs in their restaurant is really great, and they can't afford to make meatballs again and again every time their stock gets empty. They have arranged a hiring program, where you can apply with their meatball.  
They will add the meatball in their seekh (a queue) and everytime they cut the meatball they take it and cut it on the day's quantity and then re-add the meatball in the seekh. You are the hiring manager there and you are going to say who is gonna be hired.
- Day's quantity means, on that very day the company sells only that kg of meatballs to every packet.
- If someone has less than a day's quantity, it will be counted as a sell.

Name	Type	Description
N	Integer	How many people are participating in the hiring process.
D	Integer	Day's quantity, how many grams of meatball is being sold to every packet.
Array[ ]	Integer array	Array of integers, the weight of meatballs everyone came with.

**Return:**

- The ith person whose meat is served at last

**Input Format:**

First line contains N.

Second line contains D.

After that N lines contain The ith person's meatball weight.

**Output Format:** The 1 based index of the man whose meatball is served at the last.

**Sample Input 1:**

4

2

[7 8 9 3]

**Sample Output 1:**

3

**Explanation:**

The seekh or meatball queue has [7 8 9 3] this distribution. At the first serving they will cut 2 kgs of meatball from the first meatball and add it to the last of the seekh, so after 1st time it is:

[8 9 3 5]

Then, it is: [9 3 5 6], [3 5 6 7], [5 6 7 1], [6 7 1 3], [7 1 3 4], [1 3 4 5], [3 4 5], [4 5 1], [5 1 2], [1 2 3], [2 3], [3], [1], [0]

So the last served meatball belongs to the 3rd person.

## Constraints:

$$1 \leq N \leq 10^3$$

$$1 \leq D \leq 10^3$$

$$1 \leq \text{Array}[i] \leq 10^3$$





# Self Sufficient

- **Problem Statement** – Abhijeet is one of those students who tries to get his own money by part time jobs in various places to fill up the expenses for buying books. He is not placed in one place, so what he does, he tries to allocate how much the book he needs will cost, and then work to earn that much money only. He works and then buys the book respectively. Sometimes he gets more money than he needs so the money is saved for the next book. Sometimes he doesn't. In that time, if he has stored money from previous books, he can afford it, otherwise he needs money from his parents.
- Now His parents go to work and he can't contact them amid a day. You are his friend, and you have to find how much money minimum he can borrow from his parents so that he can buy all the books.
- He can Buy the book in any order.



Name	Type	Description
N	Integer	How many Books he has to buy that day.
EarnArray[ ]	Integer array	Array of his earnings for the ith book
CostArray[ ]	Integer array	Array of the actual cost of the ith book.

**Output Format:** The minimum money he needs to cover the total expense.

**Sample Input 1:**

3

[3 4 2]

[5 3 4]

**Sample Output 1:**

3

## Constraints:

$$1 \leq N \leq 10^3$$

$$1 \leq \text{EarnArray}[i] \leq 10^3$$

$$1 \leq \text{CostArray}[i] \leq 10^3$$



# Concatenation

Input: a string of comma separated numbers. The numbers 5 and 8 are present in the list Assume that 8 always comes after 5.

Case 1: num1 = add all numbers which do not lie between 5 and 8 in the input.

Case 2: num2= numbers formed by concatenating all numbers from 5 to 8.

Output: sum of num1 and num2

Example: 1) 3,2,6,5,1,4,8,9

Num1 :  $3+2+6+9=20$

Num2: 5148

output:  $5248+20=5168$

**Answer: 13**



# Unique Elements

Take input a number 'N' and an array as given below.

Input:-N = 2

Array =1,2,3,3,4,4

O/p : 2

Find the least number of unique elements after deleting N numbers of elements from the array.

In the above example, after deleting N=2 elements from the array.

In above 1,2 will be deleted.

So 3,3,4,4 will be remaining so,

2 unique elements are in the array i.e 3 and 4.





# Largest Even Number

A string which is a mixture of letters, numbers, and special characters from which produce the largest

even number from the available digit after removing the duplicates digits.

If an even number did not produce then return -1.

Ex: infosys@337

O/p : -1

.....

Hello#81@21349

O/p : 984312