**Good Elements Bad Elements**

You are given a matrix of N rows and M columns where good elements are those numbers which are even numbers and bad elements are the odd numbers. You need to find the column which has maximum good quality of elements in it.

Good quality is calculated by adding the element if it is a good element and subtract if its bad element.

**Input Format**

First line contain single integer T denoting number of test cases.

Followed by T test cases which includes:

First Line: Two integers N and M separated by spaces.

Next N lines contain M integers separated by spaces.

**Constraints**

3 <= N,M <= 10^5

**Output Format**

Single line containing the index of column and its quality.

**Sample Input 0**

1

3 4

2 4 1 3

6 3 1 4

7 4 5 2

**Sample Output 0**

1 5

**Explanation 0**

Column with index 1 has the maximum good quality i.e 4 - 3 + 4 = 5. Thus the output is 1 5

**Zuru and 2 pieces of Kaju Katli**

Zuru is a gym freak and hired a personal trainer to train him and prepare him for bodybuilding competition. He has to follow strict diet to be in a good shape and maintain his ripped physique. Zuru love to eat sweets, and can’t control himself, but the trainer takes care of this habit of him.

On Sundays, trainer allows Zuru to eat his favourite food, so as to give a break from extensive routine. Zuru’s neighbour just got married a few days ago and sent a box of sweets consisting “Kaju Katli (or Kaju ki barfi)” to him as a return gift. Zuru got reminded of it and headed to the kitchen, but the trainer had already managed to hide the box. Zuru requested the trainer to give him some pieces but the trainer didn’t want him to consume high calorie content food.

Trainer got an idea, told Zuru to write a program for the pattern that the trainer drew on a piece of paper and promised him, if the program runs successfully then he will get 2 pieces of the sweet. The trainer also added that Zuru may alter the dimension of the pattern.

The pattern drawn by the trainer is shown in sample test case.

**Input Format**

The input will contain a single integer n.

**Constraints**

1 <= n <= 1000

**Output Format**

The figure as shown in sample test case.

**Sample Input 0**

7

**Sample Output 0**

^ ^ ^ ^ ^ ^ ^^ ^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^^^^^^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^^^^^^^^^^ ^ ^ ^ ^

^ ^ ^ ^^^^^^^^^^^^^^ ^ ^ ^

^ ^ ^^^^^^^^^^^^^^^^^^ ^ ^

^ ^^^^^^^^^^^^^^^^^^^^^^ ^

^^^^^^^^^^^^^^^^^^^^^^^^^^

^^^^^^^^^^^^^^^^^^^^^^^^^^

^ ^^^^^^^^^^^^^^^^^^^^^^ ^

^ ^ ^^^^^^^^^^^^^^^^^^ ^ ^

^ ^ ^ ^^^^^^^^^^^^^^ ^ ^ ^

^ ^ ^ ^ ^^^^^^^^^^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^^^^^^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^ ^^ ^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^ ^^ ^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^^^^^^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^^^^^^^^^^ ^ ^ ^ ^

^ ^ ^ ^^^^^^^^^^^^^^ ^ ^ ^

^ ^ ^^^^^^^^^^^^^^^^^^ ^ ^

^ ^^^^^^^^^^^^^^^^^^^^^^ ^

^^^^^^^^^^^^^^^^^^^^^^^^^^

^^^^^^^^^^^^^^^^^^^^^^^^^^

^ ^^^^^^^^^^^^^^^^^^^^^^ ^

^ ^ ^^^^^^^^^^^^^^^^^^ ^ ^

^ ^ ^ ^^^^^^^^^^^^^^ ^ ^ ^

^ ^ ^ ^ ^^^^^^^^^^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^^^^^^ ^ ^ ^ ^ ^

^ ^ ^ ^ ^ ^ ^^ ^ ^ ^ ^ ^ ^

**Help The Company**

Help the company to hire three people out of a bunch of given people such that the total payment for all these people is less than the budget given.

Find out the total number of such possible groups of three that can be hired.

**Input Format**

First line has the number of test cases, n.

First line of each test case has two numbers, x and y, where x is the number of total people and y is the budget of the company.

Second line of the test case has x number of integers which is the payment required by each person.

**Constraints**

n<1000

x<10^4 and y<10^4

-10^4

**Output Format**

Output must have n number of lines, where each line must be the total number of possible groups(of three) that can be hired by the company.

**Sample Input 0**

2

5 12

3 4 1 5 7

5 13

3 4 1 5 7

**Sample Output 0**

4

6

**Explanation 0**

For the first test case, there are 4 possible groups(of three) that can be hired.

(1,3,4), (1,3,5), (1,4,5), (1,3,7)

All these groups have total less than 12.

**Nobita's Pizza**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/nobitas-pizza)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/nobitas-pizza/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/nobitas-pizza/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/nobitas-pizza/forum)

Nobita was feeling extremely hungry and ordered a giant pizza. Doraemon had a unique pizza slicer that can cut slices of areas 1, 2 or 3 only. Given the area of the giant pizza, calculate the number of ways in which Nobita can slice the pizza into different slices of areas 1, 2 and 3.

**Input Format**

A single integer value

**Constraints**

1 < area\_of\_pizza < 10^5

**Output Format**

Single Integer, number of possiblities

**Sample Input 0**

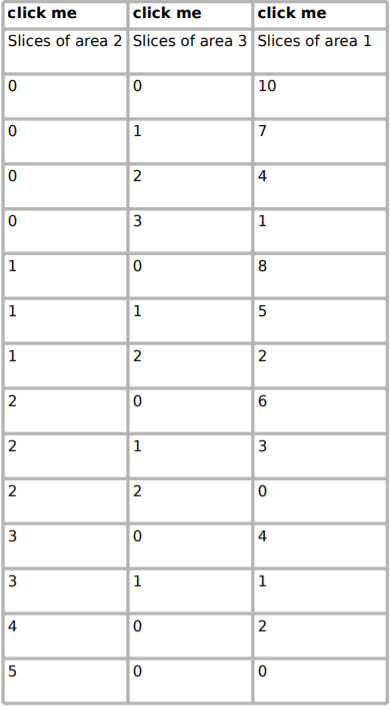
10

**Sample Output 0**

14

**Explanation 0**

The pizza with an area of 10 can be cut into slices in the following ways:



**Likhita's Bakery**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/likhitas-bakery)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/likhitas-bakery/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/likhitas-bakery/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/likhitas-bakery/forum)

Likhita has set up a new bakery, and the chief chef has listed out all the items to be provided in the menu. But Likhita felt that the menu needs to be more organised and wants to sort the menu into different lists based on some common words. Given a list of items, help Likhita group the items based on the given common words into a list of lists. Items in the list not containing any common word should be grouped into as a single list.

**Input Format**

The first line contains an integer n i.e. the number of items in the menu. The second line contains n line-separated strings, where each string is an item in the menu. The next line contains an integer m i.e. the number of common words. The next line contains n line-separated strings, where each string is a common word.

**Constraints**

1 < n < 10^5 1 < menu[i] < 10^7 1 < m < 10^5 1 < common\_words[i] < 10^7

**Output Format**

The output menu, which is a sorted list of lists of the input menu based on common words.

**Sample Input 0**

7

Sweet Corn Soup

Garlic Bread

Chicken Quiche

Hot n Sour Soup

Garlic Spaghetti

Corn Puff

Chicken Mayo Sandwich

3

Soup

Garlic

Chicken

**Sample Output 0**

[['Sweet Corn Soup', 'Hot n Sour Soup'], ['Garlic Bread', 'Garlic Spaghetti'], ['Chicken Quiche', 'Chicken Mayo Sandwich'], ['Corn Puff']]

**Explanation 0**

The items containing the word ‘soup’ have been grouped together, similarly items containing the word ‘garlic’ have been grouped together and so on, since the item 'Corn Puff’ does not have any common word, it is left as a separate group and these groups are stored in a list forming a nested lists.

**Sample Input 1**

12

Mushroom & Cheese Pasta

Tandoori Paneer Pizza

Grilled Tofu Burger

Arabiatta Pasta

Diavola Pizza

Desi Veg burger

Ginger Paneer Roll

Veg Mint Sandwich

Samosa Chaat

Paneer Lollipop

Primavera Pasta

Barbeque Pizza

6

Paneer

Pasta

Pizza

Burger

Sandwich

Veg

**Sample Output 1**

[['Tandoori Paneer Pizza', 'Ginger Paneer Roll', 'Paneer Lollipop'], ['Mushroom & Cheese Pasta', 'Arabiatta Pasta', 'Primavera Pasta'], ['Tandoori Paneer Pizza', 'Diavola Pizza', 'Barbeque Pizza'], ['Grilled Tofu Burger'], ['Veg Mint Sandwich'], ['Desi Veg burger', 'Veg Mint Sandwich'], ['Samosa Chaat']]

**Explanation 1**

All the items which have Paneer, Pasta, Pizza, Burger, Sandwich and Veg have been grouped together. Since Samosa Chaat doesn’t have a common word, so, it is stored in a separate group. Then these groups are stored in a list forming nested lists.

**Game of Balanced String**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/balanced-string-5)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/balanced-string-5/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/balanced-string-5/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/balanced-string-5/forum)

A game of letters asks you to find if a given string is balanced or not.

A string is said to be balanced if and only if the difference between the frequency of one letter with it’s adjacent letter (note: the adjacent letter is the next letter which is not same as the current letter) either the left adjacent letter or the right adjacent letter or both is exactly 1.

The game also mentions that once a letter has been considered to check if the string is balanced or not cannot be considered again as we further move into the string.

For example: In the given string ‘ZMMMO’ here the adjacent letter to the right of M is O and to the left of M is Z.

**Input Format**

First line contains T the number of test case. Second line contains a string of length l.

**Constraints**

1<=t<=10^2 2<=l<=10^4

**Output Format**

Print “Balanced” if the string is satisfies the balanced condition else print “Not balanced”

**Sample Input 0**

1

MMMOZCODEE

**Sample Output 0**

Balanced

**Explanation 0**

Frequency of M = 3, Frequency of adjacent letter O = 2, 3-2=1 (balanced condition holds)

Next , frequency of O= 2, Frequency of adjacent letter M = 3, 3-2=1(balanced condition holds)

Next, Frequency of Z=1, Frequency of adjacent letter O=2, 2-1=1(balanced condition holds)

Next, Frequency of C=1, Frequency of adjacent letter O=2, 2-1=1 (balanced condition holds)

Next, Frequency of D=1, Frequency of adjacent letter O=2, 2-1=1(Balanced condition holds)

Next, Frequency of E=2, Frequency of adjacent letter D=1, 2-1=1(balanced condition holds)

Balance condition is satisfied for all the unique letters in the string hence the string is Balanced.

**Spellers**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/spellers)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/spellers/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/spellers/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/spellers/forum)

Long long ago there was a kingdom named Winterfell. King Brandon Stark rules over Winterfell. He has habit of testing everyone’s intelligence. Whoever answers his question or solves his puzzle will be awarded with 1 bag of gold.

Once as usually a weird puzzle popped into the king’s mind. The puzzle is a sensical or nonsensical string will be given you will have to find the letters which are hidden in the string. The hidden letters will follow a logical pattern which you will have to figure out.

Lord Tyrion of House Lannister gave you a hint that it has to do something with ascii value of the letters which are present in the place values which are a multiple of 3. You figure out the rest that the sum of all those numbers are taken and if the result represents ascii value of an alphabet then the alphabet is a part of the answer else 20 is subtracted from the result and the condition is checked again.

**Input Format**

A single string is given

**Constraints**

1 <= length of string <= 10^6

**Output Format**

The correct hidden words are expected to be printed.

**Sample Input 0**

apples

**Sample Output 0**

k W C

**Explanation 0**

The ascii value of p is 112 and s is 115 so result will be 227 now 227 is not a ascii value of an alphabet 20 is subtracted and this goes on until a valid ascii value is found. 107 is the first valid ascii value so it is printed.

**Change the charge**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/change-the-charge)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/change-the-charge/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/change-the-charge/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/change-the-charge/forum)

Given an array of charges, you task is have to print the count of maximum number of +’s you can obtain by changing the charge of all elements of any one chosen sub-array at most once.

Changing the charge means to change '+' to '-' and vice versa.

If you change the charge of an sub-array, all the elements of that sub-array must change the charge.

**Input Format**

The first line of input consists of a single integer T denoting the total number of the test cases.

The first line of each test case contains an integer N, which represents the array's size.

The second line of each test case contains N space-separated integers representing the array elements accordingly.

**Constraints**

1 <= T = 100

1 <= N <= 10^4

0 <= ARR[i] <= 1

**Output Format**

For each test case, return a single integer representing the maximum number of +'s you can have in the array after at most one change operation.

**Sample Input 0**

1

5

+ + - - +

**Sample Output 0**

5

**Explanation 0**

After changing both the '-' from index 3 to 4, we get an array of {+, +, +, +, +}. Hence the output 5.

**Perfectly balanced, as all things should be**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/perfectly-balanced-as-all-things-should-be)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/perfectly-balanced-as-all-things-should-be/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/perfectly-balanced-as-all-things-should-be/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/perfectly-balanced-as-all-things-should-be/forum)

Thanos wants to balance his ideal knife but to do so he needs to identify the point about which the weight of the knife on both the sides are equal. His knife is represented by 1-D array, the elements of which represent the weight of that point. Your task is to find the index of the array about which the total weight on its left is equal to total weight on its right, in case there is no such index i.e. the knife is not ideal print -1 as output.

**Input Format**

First line contains single integer T denoting number of test cases.

Followed by T number of test cases which include:

First Line: Single Integer N denoting the number of weight points

Second Line: Weights of N points separated by spaces.

**Constraints**

1<=T<=100 1<=N<=10^4

**Output Format**

T lines, containing index of the balance point or -1

**Sample Input 0**

1

5

1 3 0 3 1

**Sample Output 0**

2

**Explanation 0**

2 is the balance point as -

A[0] + A[1] = A[3] + A[4]

**Sample Input 1**

1

6

2 5 2 2 6 3

**Sample Output 1**

3

**Explanation 1**

3 is the balance point as -

A[0] + A[1] + A[2] = A[4] + A[5] + A[6]

**Identifying the Integers**

* [**Problem**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/identifying-the-integers)
* [**Submissions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/identifying-the-integers/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/identifying-the-integers/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/mozcode21-round1/challenges/identifying-the-integers/forum)

Sam chose two integers x and y and then forgot what were the numbers, but when he selected the numbers he made a shuffled list of all the divisors of x (including 1 and x) and divisors of y (including 1 and y). If we have a number z which is a divisor of both numbers then there will be two occurences of z in the list.

Let's take a example of x = 4 and y = 10, then the given list will be [1, 2, 4, 1, 2, 5, 10] (this list can be in any permutation).

Now Sam wants you to find the positive integers x and y that would return the same list of divisors.

It is confirmed that for the given list there exists some positive integers x and y.

**Input Format**

The first line contains a single integer n which represents the number of divisors of x and y.

The second line of the input contains n integers d1, d2, ..... dn (1  di  104), here di is either divisor of x or divisor of y.

**Constraints**

2  n  128

1  di  104

**Output Format**

Print the two positive integers x and y in space-separated manner.

**Sample Input 0**

10

10 2 8 1 2 4 1 20 4 5

**Sample Output 0**

20 8

**Explanation 0**

So, the first line tells you the length of the array i.e 10.

From the array we can see that 1, 2, 4, 5, 10, 20 are the divisors of 20 and 1, 2, 4, 8 are the divisors of 8.

So the answer would be 20 and 8.