Algorithms for Programming Contests
WS18 - Week 9

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Tobias Meggendorfer, Philipp Meyer,
Christian Müller, Gregor Schwarz

Welcome to our practical course! This problem set is due by

Wednesday, 09.01.2018, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

This page is intentionally left blank.

Problem A
Goldbach's Conjecture

Goldbach's conjecture is a mathematical conjecture (that means, it is not yet proven but believed to be correct) that states the following:

Every even integer greater than 2 can be expressed as the sum of two primes.

Every odd integer greater than 5 can be expressed as the sum of three primes.

Lea is not quite convinced that this is true and asks you to prove it (and just like most computer scientists, this proof will be by example): She gives you a number and expects you to find two or three primes (depending on whether her number is even or odd) that sum up to her number.

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing one integer n, Lea's number.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space separated list of two or three primes that sum up to Lea's number. If there are multiple solutions, give the lexicographically smallest one. (I.e. if your solution is $x_1 x_2 x_3$ minimize x_1, then x_2. This in particular implies $x_1 \leq x_2 \leq x_3$.) Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $n \leq 10^8$
- n is either even and greater than 2 or odd and greater than 5

Sample Input 1	Sample Output 1
3 6 18 41	Case #1: 3 3 Case #2: 5 13 Case #3: 2 2 37

Sample Input 2	Sample Output 2
13 77 98 55 42 48 76 11 7 14 18 68 25 89	Case #1: 2 2 73 Case #2: 19 79 Case #3: 3 5 47 Case #4: 5 37 Case #5: 5 43 Case #6: 3 73 Case #7: 2 2 7 Case #8: 2 2 3 Case #9: 3 11 Case #10: 5 13 Case #11: 7 61 Case #12: 3 3 19 Case #13: 3 3 83

This page is intentionally left blank.

Problem B
Break In

In Lea's hometown, there is a building of a branch of the Innovative Consumer Products Company (ICPC). The ICPC develops many interesting products, but is very secretive about them until they are released. Lea would like to know what their next product is, so one night, she approaches the building and examines it. The building is a large black structure without any windows and only a single door. Next to the door, there are two numeric keypads, labelled with **X** and **Y**, which are connected to a digital display. Instinctively, Lea assumes that this is the key to opening the door. She tries entering several random numbers on each keypad, but the door does not open. However, she makes some interesting observations.

Firstly, after entering numbers on the keypads **X** and **Y**, the product of the two numbers is displayed on the display above. Secondly, as the display has a fixed number of digits, if the product of the numbers is too large, the remaining digits at the front are cut off. If the product is small, the remaining digits show a `0`, i.e. the number is padded with leading zeros. For example, if the display has 3 digits, and Lea enters 3 and 5, then `015` is displayed. If Lea enters 59 and 37, then `183` is displayed.

Being unsuccessful in opening the door, Lea decides to come back the following nights for further observations of the building, while hiding in nearby bushes. At certain times in the night, she notices employees of the ICPC, always in pairs, enter the building. They do this by each one of them entering a number on each keypad, respectively.

Lea cannot see the numbers entered on the keypads, however she can view the display. The number displayed there is always a 1, with leading zeros, when the door opens (e.g. `00001`). Lea is sure that the correct numbers for opening the door have to yield this result. Further, by listening to the beeps made when the numbers are entered, she knows that the number of digits entered on either keypad is at least one and never exceeds the number of digits displayed. Finally, after the persons left, through careful analysis of fingerprints and dust build up, she concluded that the last digit entered on the keypad **Y** is 1, 3, 7 or 9. The person entering numbers on keypad **X** always wears gloves, so she could not obtain any information about the number entered there.

On one lucky night, one of the two persons was not careful enough, and Lea managed to see which number was entered on keypad **Y**. Can you tell her which corresponding number needs to be entered on keypad **X** so she can open the door?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line consisting of two integers n, the number of digits on the display, and y, the number entered on the keypad **Y**.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a number with $1 \leq x \leq 10^n - 1$ such that the last n digits of the product $x \cdot y$, possibly padded with leading 0s, are a series of $n-1$ 0s followed by a single 1. You may assume that such an x always exists. If more than one such x exists, you may output any one of them.

Constraints

- $1 \leq t \leq 1000$
- $1 \leq n \leq 18$
- $1 \leq y \leq 10^n - 1$
- The last digit of y is 1, 3, 7 or 9.

Sample Input 1

```
11
1 1
1 7
1 9
2 17
2 23
2 83
2 11
3 1
3 17
3 713
3 373
```

Sample Output 1

```
Case #1: 1
Case #2: 3
Case #3: 9
Case #4: 53
Case #5: 87
Case #6: 47
Case #7: 91
Case #8: 1
Case #9: 353
Case #10: 777
Case #11: 437
```

Sample Input 2

```
12
3 969
3 589
3 923
4 333
4 7289
4 61
4 7541
4 9231
4 613
5 60543
5 11327
5 21061
```

Sample Output 2

```
Case #1: 129
Case #2: 309
Case #3: 987
Case #4: 6997
Case #5: 9609
Case #6: 7541
Case #7: 61
Case #8: 6671
Case #9: 5677
Case #10: 13407
Case #11: 23263
Case #12: 46541
```

Problem C
Cookies

Most people are quite happy to invite friends. Lea is too, and of course she strives to make the invitees as happy as possible. Sometimes, this proves to be quite difficult. This time she plans to buy cookies for everyone. While this sounds like a simple task, the eating habits of her friends complicate things.

When eating cookies, all friends sit in a big circle. All cookies are poured into a big bowl that is passed around. Each friend has a specific number of cookies that he eats every time he gets the bowl. The bowl starts full at Lea, is passed around each time in the same order, and Lea always eats exactly one cookie whenever she gets the bowl back.

She now wants to buy a number of cookies such that, no matter which of her friends show up, the bowl will end up empty after Lea takes a cookie (Then the bowl is passed around no more). It may be passed around a couple of times, but it should not happen that a friend cannot take his number of cookies or that it returns to Lea empty.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of Lea's friends. The next line contains a space separated list of n integers $c_1, \ldots, c_n$, c_i is the number of cookies her i-th friend eats each time he gets the bowl.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimal number of cookies Lea has to buy to satisfy the constraints above modulo $2147483647 = 2^{31} - 1$. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 500$
- $1 \leq c_i \leq 50$

Sample Input 1	Sample Output 1
3 2 3 5 3 1 1 1 6 45 46 47 48 49 50	Case #1: 36 Case #2: 12 Case #3: 1929057266

Sample Input 2	Sample Output 2

<pre>7
2
2 3
 	
3
2 2 2
 	
10
4 5 7 2 5 4 4 5 3 3
 	
3
4 2 2
 	
1
2
 	
10
7 1 7 5 5 6 7 3 3 4
 	
15
2 1 3 7 1 4 5 5 6 5 2 7 5 3 2</pre>
	<pre>Case #1: 12
	Case #2: 105
	Case #3: 2102860397
	Case #4: 315
	Case #5: 3
	Case #6: 1835120897
	Case #7: 1264400816</pre>

Problem D
Expired License

Paul is an extremely gifted computer scientist who just completed his master's degree at a prestigious German university. Now he would like to culminate his academic career in a PhD. The problem is that there are so many great universities out there that it is hard for him to pick the best. Because some application deadlines are coming up soon, Paul's only way to procrastinate his decision is by simply applying to all of them.

Most applications require Paul to attach a portrait photo. However, it seems like there does not exist an international standard for the aspect ratio of these kinds of photos. While most European universities ask Paul to send a photograph with aspect ratio 4.5 by 6, some Asian countries discard the applications immediately if the photo does not have an aspect ratio of 7.14 by 11.22, precisely.

As Paul has never been interested in photo editing, he never had a reason to spend a lot of money on proper software. He downloaded a free trial version some months ago, but that version has already expired and now only works with some funny restrictions. The cropping tool, for example, no longer accepts arbitrary numbers for setting the aspect ratio, but only primes. This makes Paul wonder whether the desired aspect ratios can even be properly expressed by two prime numbers. Of course, in case this is possible, he would also like to know the primes he has to enter.

Input

The first line of the input contains an integer t. t test cases follow.

The i-th test case consists of a single line containing two real numbers a_i and b_i, where $a_i \times b_i$ is the desired aspect ratio of the i-th application that Paul has to file.

Output

For each test case, output one line containing "Case #i:" with i being the number of the test case starting at 1. If it is possible to represent the desired aspect ratio by two prime numbers p and q, append the case header by p and q. Otherwise, output `impossible` after the case header. If multiple solutions exist, output the one minimizing $p + q$.

Constraints

- $1 \leq t \leq 10^5$
- $0 < a_, b_i < 100$ for all $i \in [t]$
- All real numbers are given with at most 5 decimal places after the decimal point.

Sample Input 1	Sample Output 1
3 4.5 6 7.14 11.22 0.00002 0.00007	Case #1: impossible Case #2: 7 11 Case #3: 2 7

This page is intentionally left blank.

Problem E
Absurdistanian Calendar

One of the first things Lea learnt when she first visited Absurdistan was their unusual approach to using calendars. They decided that the weekend is the best part of the week, so they extended it to three days with a new “Chillday” between Saturday and Sunday. This day is used to recover between the parties on the other days of the weekend. Thus, an absurdistanian week has eight days.

Also, they are very superstitious. The Absurdistanians believe that you have to be very careful to not cause any accident on every thirteenth of the month, no matter whether it is a Friday or any other day of the week. Skipping this day of the month is of no use since this will only result in more disasters. After this bad day there will probably not be any good day this month anymore, so they just end each month after the thirteenth.

Lea is very confused by this local calendar and while thinking about it she forgot the dates of her friend’s birthdays. She knows how many days are still left until the birthdays and needs to know on which day of the month each birthday occurs. The month itself is not important, she just wants to know for all birthdays whether it will be on a thirteenth of a month. To make things a little easier Lea wrote the numbers of days until the birthdays in base 8 to represent a full week by 10 for instance. So she sits down on this Chillday which happens to be the third of the month and asks for your help. Can you solve the problem?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing the number x of days to pass until the birthday of one of Lea’s friends in base 8.

Output

For each test case, output one line containing “Case #i: y” where i is its number, starting at 1, and y is the number of the birthday during its month. Print these days as usual, namely in base 10. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq x \leq 8^{2*10^5}$

Sample Explanation

In the first sample Lea wants to know about a day 17_8 days in the future, that is 15_{10} days base 10. Today is the third, so it takes 10 more days to finish the month. Therefore, the birthday in question is the fifth of its month.

Sample Input 1	Sample Output 1
5 17 12 13 1 12345	Case #1: 5 Case #2: 13 Case #3: 1 Case #4: 4 Case #5: 9

Sample Input 2	Sample Output 2
20 62025701 732005 6522000 6505013 3573353 43537 4510425464 442557432 552260 65672 10070270 166255113 3055763 6101546 1543116 300610251 6612345631 4561615336 610624 303335476	Case #1: 13 Case #2: 12 Case #3: 10 Case #4: 8 Case #5: 11 Case #6: 9 Case #7: 9 Case #8: 10 Case #9: 13 Case #10: 8 Case #11: 4 Case #12: 6 Case #13: 9 Case #14: 4 Case #15: 9 Case #16: 3 Case #17: 2 Case #18: 1 Case #19: 1 Case #20: 9