**The Great Computer Challenge, 2020**

***Scientific/Non-Business Programming, Level 4***

# **Challenge 1: Alphabet Cake**

You are catering a party for some children, and you are serving them a cake in the shape of a grid with **R** rows and **C** columns. Your assistant has started to decorate the cake by writing every child's initial in icing on exactly one cell of the cake. Each cell contains at most one initial, and since no two children share the same initial, no initial appears more than once on the cake.

Each child wants a single rectangular (grid-aligned) piece of cake that has their initial and no other child's initial(s). Can you find a way to assign every blank cell of the cake to one child, such that this goal is accomplished? It is guaranteed that this is always possible. There is no need to split the cake evenly among the children, and one or more of them may even get a 1-by-1 piece; this will be a valuable life lesson about unfairness.

### **Input:**

### The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each begins with one line with two integers **R** and **C**. Then, there are **R** more lines of **C** characters each, representing the cake. Each character is either an uppercase English letter (which means that your assistant has already added that letter to that cell) or ? (which means that that cell is blank).

### **Output:**

For each test case, output one line containing Case #x: and nothing else. Then output **R** more lines of **C** characters each. Your output grid must be identical to the input grid, but with *every* ? replaced with an uppercase English letter, representing that that cell appears in the slice for the child who has that initial. You may not add letters that did not originally appear in the input. In your grid, for each letter, the region formed by all the cells containing that letter must be a single grid-aligned rectangle.

If there are multiple possible answers, you may output any of them.

### **Limits:**

1 ≤ **T** ≤ 100.
There is at least one letter in the input grid.
No letter appears in more than one cell in the input grid.
It is guaranteed that at least one answer exists for each test case.

### **Dataset:**

1 ≤ **R** ≤ 12.
1 ≤ **C** ≤ 12.
**R** × **C** ≤ 12.

### **Sample**



The sample output displays one set of answers to the sample cases. Other answers may be possible.

# **Challenge 2: Foregone Solutions**

Someone just won the lottery, and we owe them **N** coins! However, when we tried to print out an oversized check, we encountered a problem. The value of **N**, which is an integer, includes at least one digit that is a 4... and the 4 key on the keyboard of our oversized check printer is broken.

Fortunately, we have a workaround: we will send our winner two checks for positive integer amounts A and B, such that neither A nor B contains any digit that is a 4, and A + B = **N**. Please help us find any pair of values A and B that satisfy these conditions.

### **Input:**

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each consists of one line with an integer **N**.

### **Output:**

For each test case, output one line containing A, B. A and B are positive integers as described above.

It is guaranteed that at least one solution exists. If there are multiple solutions, you may output any one of them.

### **Limits:**

1 ≤ **T** ≤ 100.
At least one of the digits of **N** is a 4.

### **Test set:**

1 < **N** < 105.

### **Sample:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 |  |
| 4 | 2, 2 |
| 940 | 852, 88 |
| 4444 | 667, 3777 |

# **Challenge 3: Letter Counts**

If the numbers 1 to 5 are written out in words: one, two, three, four, five, then there are 3 + 3 + 5 + 4 + 4 = 19 letters used in total.

If all the numbers from 1 to 1000 (one thousand) inclusive were written out in words, how many letters would be used?

**NOTE:** Do not count spaces or hyphens. For example, 342 (three hundred and forty-two) contains 23 letters and 115 (one hundred and fifteen) contains 20 letters. The use of "and" when writing out numbers is in compliance with British usage.

# **Challenge 4: Theme Park**

Roller coasters are so much fun! It seems like everybody who visits the theme park wants to ride the roller coaster. Some people go alone; other people go in groups, and don't want to board the roller coaster unless they can all go together. And everyone who rides the roller coaster wants to ride again. A ride costs 1 Euro per person; your job is to figure out how much money the roller coaster will make today.

The roller coaster can hold **k** people at once. People queue for it in groups. Groups board the roller coaster, one at a time, until there are no more groups left or there is no room for the next group; then the roller coaster goes, whether it's full or not. Once the ride is over, all of its passengers re-queue in the same order. The roller coaster will run **R** times in a day.

For example, suppose **R**=4, **k**=6, and there are four groups of people with sizes: 1, 4, 2, 1. The first time the roller coaster goes, the first two groups [1, 4] will ride, leaving an empty seat (the group of 2 won't fit, and the group of 1 can't go ahead of them). Then they'll go to the back of the queue, which now looks like 2, 1, 1, 4. The second time, the coaster will hold 4 people: [2, 1, 1]. Now the queue looks like 4, 2, 1, 1. The third time, it will hold 6 people: [4, 2]. Now the queue looks like [1, 1, 4, 2]. Finally, it will hold 6 people: [1, 1, 4]. The roller coaster has made a total of 21 Euros!

### **Input:**

The first line of the input gives the number of test cases, **T**. **T** test cases follow, with each test case consisting of two lines. The first line contains three space-separated integers: **R**, **k** and **N**. The second line contains **N** space-separated integers **gi**, each of which is the size of a group that wants to ride. **g0** is the size of the first group, **g1** is the size of the second group, etc.

### **Output:**

For each test case, output one line containing "Case x: y", where x is the case number (starting from 1) and y is the number of Euros made by the roller coaster.

**Limits:**

1 ≤ **T** ≤ 50.
**gi** ≤ **k**.

### **Dataset:**

1 ≤ **R** ≤ 1000.
1 ≤ **k** ≤ 100.
1 ≤ **N** ≤ 10.
1 ≤ **gi** ≤ 10.

**Sample:**



# **Challenge 5: Constructing a number**

Manipulating numbers is at the core of a programmer's job. To test how well you know their properties, you are asked to solve the following problem.

You are given **n** non-negative integers **a1**, **a2**, ..., **an**. You want to know whether it's possible to construct a new integer using all the digits of these numbers such that it would be divisible by 3. You can reorder the digits as you want. The resulting number can contain leading zeros.

For example, consider the numbers 50, 40, 90 from which you have to construct a new integer as described above. Numerous arrangements of digits are possible; but we have illustrated one below.



Your code must take an integer array as input and return at least one constructed integer can be constructed. If no such integer is possible, return ‘NONE’.

### **Input Format**

The first line contains a single integer **t** denoting the number of queries. The following lines describe the queries.

Each query is described in two lines. The first of these lines contains a single integer **n**. The second contains **n** space-separated integers **a1, a2, ……. an.**

### **Constraints**

****

### **Output Format**

For each query, return at least one constructed number, if it exists. If there are multiple possible answers, you may output any of them. If such a number does not exist, return ‘NONE’.

**Input Output**

3

40 50 90 005490

1 4 NONE

9 9

***Have fun and thanks for participating in the
Great Computer Challenge, 2020!***