ACM-ICPC Thailand Southern Programming Contest 2013

Hosted by
Department of Computer Engineering
Prince of Songkla University Hatyai Campus

10 August 2013

Contest Problems

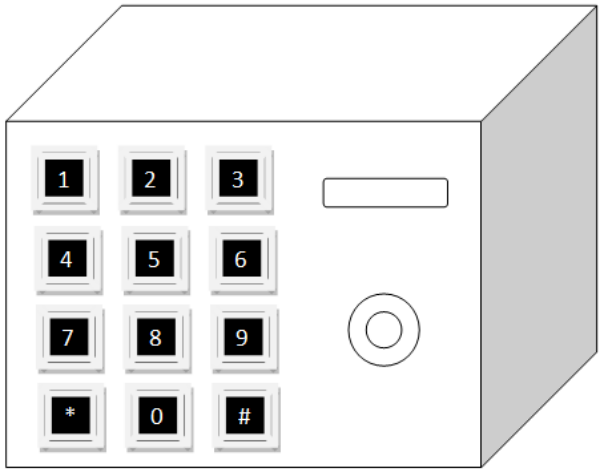
- There are **8** problems (A-H) to solve within 3 hours 30 minutes.
- Solve as many problems as you can, in an order of your choice.
- Use C or C++ or Java to program at your convenience for any problems.
- Input and output of each program are **standard input** and **output**.

Problem A	Unlock My Safe
Problem B	Two Mysterious Alphabets from a Tree
Problem C	Max Volume
Problem D	Birthday Statistics
Problem E	Nonogram
Problem F	Jane's First Words
Problem G	Range Sum Query
Problem H	Sum of Distinct Numbers ผลรวมเลขไม่ซ้ำ

Problem A. Unlock My Safe

Time Limit: 2s

I forgot the password to my safe. There is a lot of money in it! Please help me unlock the safe. The keypad looks like this.



I do not remember how long my password is. Hence, you need to try a different length of the password. However, there are some hints that I can recall.

- I never use characters *, #, 0 and 9 in my password.
- Each digit in the password is distinct. That is, they never appear more than once.
- My password is at most 8 digits (1 <= ***N*** <= 8, where ***N*** is a number of digits in the password).
- Each digit *i* in the password always has the value less than or equal to ***N*** (that is, a password 132 is valid for ***N*** = 3 but a password such as 124 is invalid because the 3rd digit exceeds 3).

Use the information above and generate all possible permutations. One permutation corresponds to one guess of a password to unlock my safe. Importantly, the correct password is deliberately fixed at position ***L***\3 in the sorted array of permutations, where ***L*** is a number of all possible permutations and '\' is an *integer division*. The sorted array of permutations is in ascending order and the starting index in the sorted array begins at 0 (not 1).

Write a program to find a correct password for a given length (a number of digits in the password).

Input

The first line of the input contains an integer ***T*** (1 <= ***T*** <= 6) denoting the number of test cases. After that ***T*** test cases follow. Each test case contains an integer ***N*** (1 <= ***N*** <= 8) denoting a number of digits in a password.

Output

Your program should output the *N*-digit password for each corresponding test case, one password per line.

Sample input	Sample output
3 2 3 1	12 213 1

Explanation

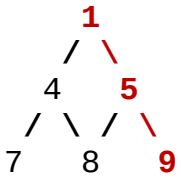
There are 3 test cases above. In the second case, for example, the sorted permutations are {123, 132, 213, 231, 312, 321}. Password is located at the position 6\3 = 2 (integer division). When the starting index begins at 0, the password is, therefore, 213.

Problem B. Two Mysterious Alphabets from a Tree

Time Limit: 2s

Your task is to extract 2 alphabets from a **binary** tree which is composed of unsigned integers respecting the following rules. Let *n* be the height of a tree. At the level *k* (1<=*k*<=*n*), the tree contains *k* of nodes and each node has 2 children nodes (except the leaf nodes at the level *n* which have no children). See the example below to understand the tree formation. Some nodes may have 2 parent nodes.

Example:



You need to walk in a tree on the path that has a maximum summation (e.g., 1 + 5 + 9 = 15). Numbers in each summation cannot cross into different links (e.g., 5+7 is illegal). Then, your intermediate task is to calculate *2 numbers* for alphabet extraction. *The first number* is calculated from $\sum_{i=1}^{n} i^2$ where *i* is a number along the maximum summation path and *n* is the height of a tree. *The second number* is a summation of the maximum path ($\sum_{i=1}^{n} i$). Regarding to the example above, the first number = 1 + 25 + 81 = 107 and the second number = 1 + 5 + 9 = 15.

Finally, these two numbers are transformed into two lower case alphabets from 'a' to 'z' respectively, where 'a' is used for 0 and 'z' is used for 25. Since there are only 26 alphabets, a number greater than 25 will reuse the same set of alphabets. For example, 107 = 'd' and 15 = 'p' (that is, the first alphabet 'a' = 0, or 26, or 52 etc).

Write a program to find the 2 mysterious alphabets from a given tree.

Input

The first line of input contains the height (*n*) of a tree (0<*n*< 100). The second line contains unsigned integer numbers (*i*) in each level of a tree (0<*i*<100), consecutively. *Assume that there is only one maximum path in a tree.*

Output

The first line contains two integer calculated from the rules above, and the second line contains 2 decoded alphabets.

Sample Input	Sample Output
3 1 4 5 7 8 9	107 15 dp

Sample Input	Sample Output
4 1 5 2 5 1 9 3 4 20 1	486 32 sg

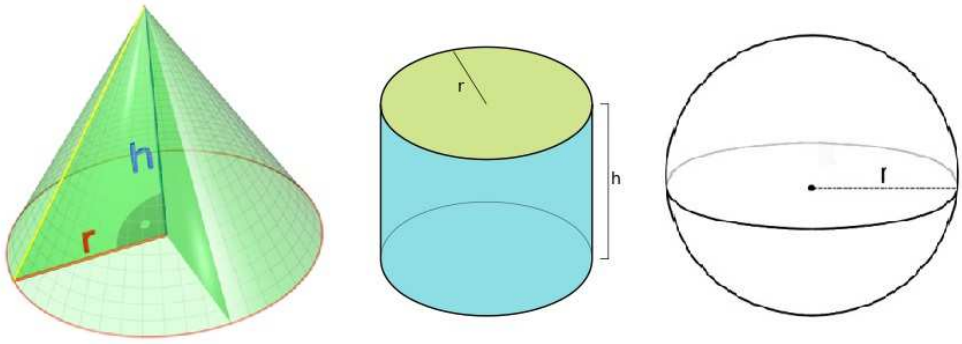
Sample Input	Sample Output
5 2 4 9 1 3 1 1 1 1 2 12 5 4 3 2	166 20 ku

Sample Input	Sample Output
6 9 8 8 7 7 7 9 1 1 3 8 2 10 5 1 2 3 2 1 9 81	6765 109 ff

Problem C. Max Volume

Time Limit: 1s

Write a program to find the maximum volume of given geometric 3-dimensional figures. Here, there are 3 types of figures: cone, cylinder and sphere.



The volume (V) of each figure can be calculated by the following formulas.

Cone: $V = (1/3)\pi r^2 h$

Cylinder: $V = \pi r^2 h$

Sphere: $V = (4/3)\pi r^3$

Use the value π **= 3.14159** in your calculation.

Input

The first line of the input contains a positive integer *n (1<=n<=100)* which is the number of figures. The *n* following lines contain the description of each figure. In case of a cone, the line begins with letter `C` and followed by 2 values: *r* and *h* respectively. If it is a cylinder, the line begins with letter `L` and followed by 2 values: *r* and *h* respectively. If it is a sphere, the line begins with letter `S` and followed by only one value which is *r*.

Output

Print out the max volume among the input figures with 3 decimal places

Sample Input	Sample Output
5 S 3.0 C 2.5 3 S 1.79 L 2.78 1.4 C 1.15 2.36	113.097

Problem D. Birthday Statistics

Time Limit: 1s

In a company, the birthdays of all employees are collected. Your task is to write a program to summarize the number of people that were born in each month.

Input

The first line of the input contains a positive integer *n (1<=n<=5,000)* which is the number of employees. The *n* following lines contain the ID (4-digit number) of each employee and his/her birth date. The birth date is written in this particular format: `day/month/year.`
Suppose that all dates in the input are valid dates.

Output

Print out the number of month (1-12) and the number of employees born in that month.

Sample Input	Sample Output
10 1000 1/2/1967 1012 13/10/1940 1103 5/1/1965 2012 16/7/1980 1125 18/9/1979 1235 10/10/1976 1400 16/11/1973 1013 5/1/1965 2109 28/7/1958 2155 10/12/1970	1 2 2 1 3 0 4 0 5 0 6 0 7 2 8 0 9 1 10 2 11 1 12 1

Problem E. Nonogram

Time Limit: 3s

Nonogram is a logic puzzle with simple rules and challenging solutions. The rules are simple. You have a grid of squares, which must be either filled in black (we denote this with 1) or marked with X/blank (we denote this with 0). Beside each row of the grid are listed the lengths of the runs of black squares on that row. Above each column are listed the lengths of the runs of black squares in that column. Your aim is to find all black squares.

Input

The first line of input contains a positive integer **TC**, the number of nonogram puzzles that you have to solve. Each test case starts with a blank line, followed by two positive integers **W** and **H** in the next line that denote the width and the height of the puzzle. We guarantee that **$1 \leq W*H \leq 20$**.

Then there will be two blocks of puzzle description that started with a blank line: one for rows (that contains **H** lines) and another for columns (that contains **W** lines). In those two blocks, you will be given comma separated values that denote the lengths of the runs of black squares in each row/columns (note: it can be blank).

Output

For each test case (puzzle), determine if the answer is unique. If it is unique, output a 2D binary matrix of **H** lin es containing **W** characters. Put 1 if the corresponding cell is black, or 0 otherwise. If it is not unique, just print "*not unique*" in one line (without "). Separate the output of two test cases with a blank line.

Sample Input

```
3

5 4

3
1,2
1,1
2

3
1
1
1,1
3

7 2

1,1,1,1
1,1,1

1
1
1
1
1
1
1
```

```
7 1

1,3,1

1

1
1
1

1
```

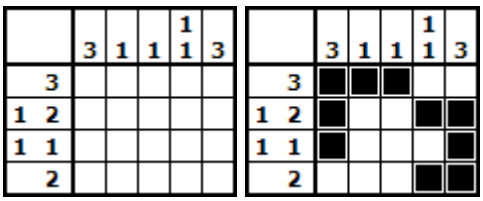
Sample Output

```
11100
10011
10001
00011

1010101
0101010

1011101
```

Visualization of the first Sample Input & Sample Output



Problem Author

Dr Steven Halim, School of Computing, National University of Singapore

Problem F. Jane's First Words

Time Limit: 1s

Problem Description

Jane (my ~2 years-old baby daughter) has started speaking some simple words now. "*Daddy*" and "*Mommy*" are the two common first words. Hearing those words for the first time are indeed beautiful and memorable.

Last year, Steven really wanted to record when his baby called him for the first time. So, Steven put a microphone and sound capture program near Jane's baby cot (baby's bed). This microphone captured Jane's sounds and the program transmitted the list of words captured to Steven. He wrote a program to detect the moment when Jane's first call him: "daddy" (**or its variants**). This time, you will also write similar program.

Input

You are given one word per line. These are the list of captured sounds.
Each line contains only lowercase alphabet without any whitespaces and at most 20 characters.
Input is terminated with an EOF.

Output

For each word/line, output **"She called me!!!"** in one line if that word matches this regular expression below. Otherwise, output **"Cooing"** in one line (to say that this is just some baby soft murmuring sound).
Note: Quotes are only for clarity.

The regular expression (regex): **"da+dd?(i|y)"**.
If you are not familiar with regex, let me explain:

'+' means *one or more* of the preceding element.
'?' means *zero or one* of the preceding element.
A vertical bar '|' separates alternatives.
Parentheses are used to define the scope and precedence of the operators.

Sample Input	Sample Output
aaaa eeeh auwww dda daaada daddy ouuuww dadi	Cooing Cooing Cooing Cooing Cooing She called me!!! Cooing She called me!!!

Problem Author

Dr Steven Halim, School of Computing, National University of Singapore

Problem G. Range Sum Query

Time Limit: 1s

Problem Description

Given a list **L** containing **n** integers, find the Range Sum Query (RSQ) between index **i** and **j**, inclusive, i.e. **RSQ(i, j) = L[i] + L[i+1] + L[i+2] + ... + L[j]**.

Input

The input starts with an integer **t** in the first line that denotes the number of test cases in this problem ($1 \leq \mathbf{t} \leq 5$).
Each test case starts with a blank line, followed by a line that contains 2 integers: **n** and **q** ($1 \leq \mathbf{n}, \mathbf{q} \leq 100{,}000$).
Then, the next line contains **n** non-negative integers up to 1,000,000,000.
Then **q** lines follow.
Each line contains two integers, **i** and **j** ($0 \leq \mathbf{i}, \mathbf{j} < 10{,}000$).

Output

For each query, print a line containing the value of RSQ(i, j).
Separate two test cases with a blank line.

Sample Input

```
2

5 2
1 2 3 4 5
4 4
1 3

10 5
10 9 7 20 14 23 14 27 38 77
8 9
7 9
6 9
5 9
4 9
```

Sample Output

```
5
9

115
142
156
179
193
```

Problem Author

Dr Steven Halim, School of Computing, National University of Singapore

Problem H. Sum of Distinct Numbers

ผลบวกเลขไม่ซ้ำ

Time Limit: 1s

ให้จำนวนเต็ม N (1 <= N <= 2,000) อยากทราบว่า N สามารถเขียนเป็นผลบวกของจำนวนเต็มที่มีค่าไม่ซ้ำกันเลยได้กี่แบบ

ตัวอย่างเช่น ถ้า N=5 เราสามารถเขียน N ได้ 3 แบบ คือ 5, 2+3 และ 1+4 สังเกตว่าเราจะไม่นับ 1+1+3 เพราะว่ามีการใช้ 1 ซ้ำกัน นอกจากนี้ เรายังพิจารณาว่า 2+3 กับ 3+2 นั้นเป็นวิธีเขียนเดียวกันด้วย (นั่นคือจะไม่นับซ้ำ)

พิจารณาตัวอย่างที่สอง เมื่อ N=6 เราสามารถเขียน N ได้ 4 แบบคือ 6, 1+5, 1+2+3 และ 2+4

Input ข้อมูลนำเข้า

บรรทัดแรกระบุจำนวนเต็ม T แทนจำนวนข้อมูลทดสอบ (1 <= T <= 20) จากนั้นอีก T บรรทัดจะระบุข้อมูลทดสอบแต่ละชุด ชุดละหนึ่งบรรทัด ข้อมูลทดสอบแต่ละชุดจะระบุจำนวนเต็ม N หนึ่งตัว

Output ข้อมูลส่งออก

สำหรับแต่ละข้อมูลชุดทดสอบ ให้พิมพ์จำนวนวิธีที่สามารถเขียน N ให้เป็นผลบวกที่แต่ละพจน์มีค่าไม่ซ้ำกัน เนื่องจากคำตอบอาจมีค่ามาก ให้พิมพ์คำตอบ modulo 100999

ตัวอย่าง

Sample Input	Sample Output
4 5 6 10 200	3 4 10 50568

Description in English

A positive integer N can be written in the form of sum of ***distinct*** positive integers in several ways. For example,

N=5, there are 3 ways: 5, 2+3, 1+4

N=6, there are 4 ways: 6, 1+5, 1+2+3, 2+4

Here the permutation of the same elements is counted as one i.e. 1+2+3 is the same as 2+1+3 and 3+1+2, etc.

Input

You are given the number of test cases (1 <= T <= 20) in the first line. Then, in the following T lines, each line contains the number N (1 <= N <= 2,000).

Output

Print out, for each number N, the number of possible ways of writing that number in the form of sum of distinct numbers as described above. In order to limit the range of answers, the answer must be the result value modulo 100999.