

Bronze 4

1 - a ten-part system

Time limit	Memory limit
1 Sec	256 MB

Problem

Starting June 1, Seoul will implement a 10-day license plate-based driving restriction system to prevent traffic congestion. This system prohibits vehicles from operating on days when the last digit of their license plate number matches the last digit of the date. For example, if the last digit of a license plate is 7, the vehicle cannot be driven on the 7th, 17th, or 27th. Additionally, if the last digit of the license plate is 0, the vehicle cannot be driven on the 10th, 20th, or 30th. You will volunteer as a one-day police officer to count the number of vehicles that violate this restriction. Given the last digit of the date and the last digits of five license plates, you must output the number of violating vehicles.

Input

The first line contains the last digit of the date, and the second line contains the last digits of the license plate numbers of five cars. Both the date and license plate digits are integers from 0 to 9 inclusive.

Output

Using the given date and the last digits of the license plate numbers, output the number of vehicles that violate the 10-day license plate-based driving restriction system.

Enter Example 1

```
1  
1 2 3 4 5
```

Example Output 1

```
1
```

Enter Example 2

```
3  
1 2 3 5 3
```

Example Output 2

Enter Example 3

```
5
1 3 0 7 4
```

Example Output 3

```
0
```

青铜 4

1 - 十部制

时间限制	内存限制
1 Sec	256 MB

问题

自 6 月 1 日起，首尔市为防止交通拥堵，将实行以车牌尾号为基础的 10 日轮换限行制度。在该制度下，如果车辆号牌的个位数字与日期的个位数字相同，则该车辆当天禁止上路行驶。例如，如果车牌号的个位数是 7，那么在 7 日、17 日和 27 日该车不得上路行驶。另外，如果车牌号的个位数字是 0，则在 10 日、20 日和 30 日不得上路行驶。你将作为“一日警察”参加志愿活动，负责统计违反该限行规定的车辆数量。给定日期的个位数字以及 5 辆汽车车牌号的个位数字，要求输出违反规定的车辆数。

输入

第一行给出日期中日的个位数字，第二行给出 5 辆车牌号的个位数字。日期中日的个位数字和车牌号的个位数字都是 0 到 9 之间的整数（包含 0 和 9）。

输出

根据给定的日期和车牌号的个位数字，输出违反该 10 日车牌尾号限行制度的车辆数量。根据给定的日期和车牌号的个位数字，输出违反该 10 日车牌尾号限行制度的车辆数量。

输入示例 1

```
1  
1 2 3 4 5
```

示例输出 1

```
1
```

输入示例 2

```
3  
1 2 3 5 3
```

示例输出 2

```
2
```

输入示例 3

```
5
1 3 0 7 4
```

示例输出 3

```
0
```

Bronze 4

2 - a square number

Time limit

Memory limit

1 Sec

1024 MB

Problem

A number that is the square of a positive integer is called a perfect square. Given a positive integer N , determine whether N is a perfect square.

Input

: The first line contains the number of test cases, T . From the second line onward, a positive integer N is given on each of the next T lines.

Restriction

$1 \leq T \leq 999$, $1 \leq N \leq 10^9$

Output

For each of the T test cases, output 1 if N is a perfect square otherwise, output 0

Enter Example 1

```
10
1
2
3
4
5
6
7
8
9
10
```

Example Output 1

```
1
0
0
1
```

```
0  
0  
0  
0  
1  
0
```

Enter Example 2

```
6  
11  
12  
13  
14  
15  
16
```

Example Output 2

```
0  
0  
0  
0  
0  
1
```

青铜 4

2 - 乘方数

时间限制	内存限制
1 Sec	1024 MB

问题

正整数的平方叫做完全平方数。给定一个正整数 N ，判断 N 是否为完全平方数

输入

第一行给出测试用例的个数 T 。从第二行开始的接下来 T 行中，每行给出一个正整数 N

限制

$1 \leq T \leq 999$, $1 \leq N \leq 10^9$

输出

对于每个测试用例，若 N 是完全平方数则输出 1，否则输出 0，共输出 T 行

输入示例 1

```
10
1
2
3
4
5
6
7
8
9
```

10 示例输出 1

```
1
0
0
1
```

```
0  
0  
0  
0  
1  
0
```

输入示例 2

```
6  
11  
12  
13  
14  
15  
16
```

示例输出 2

```
0  
0  
0  
0  
0  
1
```

Bronze 2

3 - the difference between five and six

Time limit	Memory limit
1 Sec	128 MB

Problem

Sanggeun worked so hard rotating the table in problem 2863 that he started to get confused between the digits 5 and 6. When Sanggeun sees the digit 5, he might read it as 5, but he might also mistakenly read it as 6. Likewise, when he sees the digit 6, he might read it as 6, but he might also mistakenly read it as 5. Given two numbers A and B, Sanggeun wants to add them. Write a program that, among all possible sums he could obtain, computes and prints the minimum and maximum values. Sanggeun worked so hard rotating the table in problem 2863 that he started to get

confused between the digits 5 and 6.

When Sanggeun sees the digit 5, he might read it as 5, but he might also mistakenly read it as

6. Likewise, when he sees the digit 6, he might read it as 6, but he might also mistakenly read it as 5.

Given two numbers A and B, Sanggeun wants to add them. Write a program that, among all possible sums he could obtain, computes and prints the minimum and maximum values..

Input

The first line contains two integers, A and B. ($1 \leq A, B \leq 1,000,000$).

Output

On the first line, output the minimum and maximum values of the sum of the two numbers Sanggeun can obtain.

Enter Example 1

```
11 25
```

Example Output 1

```
36 37
```

Enter Example 2

1430 4862

Example Output 2

6282 6292

Enter Example 3

16796 58786

Example Output 3

74580 85582

青铜 2

3 - 五与六之差

时间限制	内存限制
1 Sec	128 MB

问题

尚根在做第 2863 题时把表格转得太投入了，结果开始把数字 5 和 6 搞混了。当尚根看到数字 5 时，他有时会把它看成 5，但也有可能错看成 6。同样地，当他看到数字 6 时，有时会把它看成 6，也有可能错看成 5。给定两个数 A 和 B，尚根想把它们相加。在所有他可能算出来的和中，请你写一个程序，输出其中可能的最 小值和最大值

输入

第一行包含两个整数 A 和 B。 $(1 \leq A, B \leq 1,000,000)$

输出

在第一行输出尚根可能得到的两数之和的最小值和最大值

输入示例 1

```
11 25
```

示例输出 1

```
36 37
```

输入示例 2

```
1430 4862
```

示例输出 2

```
6282 6292
```

输入示例 2

```
16796 58786
```

示例输出 2

```
74580 85582
```

Bronze 2

4 - Plus

Time limit	Memory limit
1 Sec	256 MB

Problem

You are given a string S consisting only of digits and commas. Write a program to find the sum of the nature integers contained in S.

The first and last characters of S are always numbers, and commas are never given consecutively. The given numbers are always natural numbers.

Input

The first line contains a string S. The length of S is at most 100. The integers contained within are natural numbers less than or equal to 1,000,000

Output

Output the sum of the natural numbers contained in the string S.

Enter Example 1

```
10,20,30,50,100
```

Example Output 1

```
210
```

青铜 2

4 - 加

时间限制	内存限制
1 Sec	256 MB

问题

给定一个仅由数字和逗号组成的字符串 S。请编写程序，求出 S 中所包含的所有自然数之和。字符串 S 的第一个字符和最后一个字符一定是数字，并且不会出现连续的逗号。给出的所有数都是自然数

输入

第一行给出一个字符串 S。S 的长度最多为 100。S 中包含的整数都是不超过 1,000,000 的自然数

输出

输出字符串 S 中所包含的自然数之和

输入示例 1

```
10,20,30,50,100
```

示例输出 1

```
210
```

Bronze 2

5 - command prompt

Time limit	Memory limit
2 Sec	128 MB

Problem

Click Start → Run and type cmd. A black window will appear. If you type dir there, all subdirectories and files in that directory are listed. To find only the files you want, you can do the following. If you type dir *.exe, all files whose extension is exe are displayed. If you type dir pattern, only the files that match that pattern appear in the search results. For example, if you search with dir a?b.exe, all files whose first character of the file name is a, whose third character is b, and whose extension is exe are listed. The second character can be anything; for instance, acb.exe, aab.exe, and apb.exe In this problem, you are given the search results first, and you must output what pattern you would have to type in order to obtain exactly those results. A pattern may contain only Alphabet, ".", and "?". You should use as few ? characters as possible. Assume that the directory contains only the files that appear in the search results, and that all file names have the same length.

Input

The first line contains the number of file names, N. From the second line, each of the next N lines contains a file name. N is a natural number less than or equal to 50. All file names have the same length, and the length is at most 50. Each file name consists only of lowercase letters and the character '.'.

Output

Just output the pattern on the first line. Just output the pattern on the first line.

Enter Example 1

```
3
config.sys
config.inf
configures
```

Example Output 1

```
config????
```

Enter Example 2

```
2  
contest.txt  
context.txt
```

Example Output 2

```
conte?t.txt
```

Enter Example 3

```
3  
c.user.mike.programs  
c.user.nike.programs  
c.user.rice.programs
```

Example Output 3

```
c.user.?i?e.programs
```

Enter Example 4

```
4  
a  
a  
b  
b  
b
```

Example Output 4

```
?
```

Enter Example 5

```
1  
onlyonefile
```

Example Output 5

```
onlyonefile
```

青铜 2

5 - 命令提示符

时间限制	内存限制
2 Sec	128 MB

问题

点击“开始→运行”，输入 cmd，会出现一个黑色窗口。在那里输入 dir，即可列出该目录

下的所有子目录和文件。若只想查找特定文件，可以按如下方式操作。输入 dir*.exe 时，会显示所有扩展名为 exe 的文件。输入 dir 模式时，只会在搜索结果中显示与该模式匹配的文件。比如 输入 dir a?b.exe，则会列出所有文件名第一个字符为 a、第三个字符为 b、扩展名为 exe 的文件。第二个字符可以是任意字符，例如 acb.exe、aab.exe、apb.exe 等。本题中，先给出的是搜索结果，你需要输出一个模式，使得在该目录下用这个模式搜索时，能恰好得到这些文件。模式中只能使用字母 . 和 ?，并且要尽可能少使用?。假设该目录中只存在这些搜索结果中的文件，且所有文件名的长度都相同

输入

第一行给出文件名的个数 N。从第二行开始的接下来的 N 行中，每行给出一个文件名。N 是不超过 50 的自然数。所有文件名的长度都相同，且长度最多为 50。每个文件名只由小写字母和字符'.'组成

输出

只需在第一行输出该模式

输入示例 1

```
3
config.sys
config.inf
configures
```

示例输出 1

```
config????
```

输入示例 2

```
2  
contest.txt  
context.txt
```

示例输出 2

```
conte?t.txt
```

输入示例 3

```
3  
c.user.mike.programs  
c.user.nike.programs  
c.user.rice.programs
```

示例输出 3

```
c.user.?i?e.programs
```

输入示例 4

```
4  
a  
a  
b  
b
```

示例输出 4

```
?
```

输入示例 5

```
1  
onlyonefile
```

示例输出 5

```
onlyonefile
```

Silver 5

6 - Data structures

Time limit	Memory limit
1 Sec	512 MB

Problem

Ims is looking for people to play mini-games with. The available mini-games are Yutnori (Y), Picture Matching (F), and One Card (O). These games are played with 2, 3, and 4 players respectively, and the game cannot start if there are not enough players. Given the number N of times people have requested to play with Imse and the types of games Imse can play, Determine the maximum number of times Ims can play the game. Some people may want to play mini-games with Ims multiple times, but Ims does not play more than once with the same person. Among the people who want to play with Ims, there are no duplicate names. "Ims" and "Ims0806" are considered different individuals.

Input

The first line contains the number N of requests from people to play with Imse, along with the type of game to be played. ($1 \leq N \leq 1,000,000$) From the second line onward, N lines contain the names of people who wish to play with Imse, given as strings. ($1 \leq \text{String Length} \leq 20$) People's names consist of numbers or uppercase and lowercase letters

Output

Find the maximum number of times Imse can play the game.

Enter Example 1

```
7 Y

lms0806
lms0806
exponential
e lms0806
jthis
lms0806
leo020630
```

Example Output 1

```
4
```

Enter Example 2

12 F

```
lms0806
powergee
skeep194
lms0806
tony9402
lms0806
wider93
lms0806
mageek2guanaah
lms0806 jthis
lms0806
```

Example Output 2

3

Enter Example 3

```
12 O
lms0806
mageek2guanaah
jthis
lms0806
exponentiale
lms0806
leo020630
lms0806
powergee
lms0806
skeep194
lms0806
```

Example Output 3

2

银级 5

6 - 数据结构

时间限制	内存限制
1 Sec	512 MB

问题

Ims 正在寻找一起玩小游戏的伙伴。可选的小游戏有：掷栖游戏 Yutnori (Y)、找相同图案 Picture Matching (F) 和 One Card (O)。它们分别需要 2、3、4 名玩家参与，如果人数不足，则无法开始游戏。给定人们向 Ims 申请一起玩游戏的次数 N 以及 Ims 要玩的游戏类型，求 Ims 最多可以和别人一起玩多少次游戏。有些人可能想和 Ims 多次玩小游戏，但 Ims 不会和同一个人一起玩超过一次。在所有想和 Ims 一起玩游戏的人当中，不存在同名的人。“Ims”和“Ims0806”被视为不同的个人。

输入

第一行给出申请与 Ims 一起玩游戏的次数 N 以及要进行的游戏类型。 $(1 \leq N \leq 1,000,000)$

从第二行开始的 N 行中，给出了希望共同游戏的人们的名字，以字符串形式呈现。 $(1 \leq \text{字符串长度} \leq 20)$

人们的名字由数字或英文字母（大小写）组成

输出

求 Ims 最多可以进行游戏的次数

输入示例 1

```
7 Y

lms0806
lms0806
exponential
e lms0806
jthis
lms0806
leo020630
```

示例输出 1

输入示例 2

```
12 F

lms0806
powergee
skeep194
lms0806
tony9402
lms0806
wider93
lms0806
mageek2guanaah
lms0806 jthis
lms0806
```

示例输出 2

```
3
```

输入示例 3

```
12 O

lms0806
mageek2guanaah
jthis
lms0806
exponentiale
lms0806
leo020630
lms0806
powergee
lms0806
skeep194
lms0806
```

示例输出 3

```
2`
```

Silver4

7 - Josephus problem

Time limit	Memory limit
2 Sec	256 MB

Problem

The Josephus problem is defined as follows. N people, numbered from 1 to N, are sitting in a circle. A positive integer K ($\leq N$) is given. Starting from the first person, you repeatedly remove every K-th person in order. After a person is removed, the procedure continues along the circle with the remaining people. This process is repeated until all N people have been removed. The order in which people are removed from the circle is called the (N, K)-Josephus permutation. For example, the (7, 3)-Josephus permutation is (3, 6, 2, 7, 5, 1, 4). Given N and K, write a program to compute the (N, K)-Josephus permutation.

Input

The first line contains N and K separated by a space. ($1 \leq K \leq N \leq 5,000$)

Output

Output the Josephus permutation as shown in the example.

Enter Example 1

```
7 3
```

Example Output 1

```
<3, 6, 2, 7, 5, 1, 4>
```

银级 4

7 - 约瑟夫斯问题

时间限制 内存限制

2 Sec 256 MB

问题

约瑟夫问题定义如下：有 N 个人，按顺序从 1 到 N 编号，围成一个圆圈而坐。给定一个正整数 K ($K \leq N$)。从第 1 个人开始，按照顺时针顺序每数到第 K 个人时，将其从圆圈中移除。某个人被移除后，从下一个人开始继续同样的过程。这个过程一直重复，直到 N 个人全部被移除为止。人们被依次移除的顺序称为 (N, K) -约瑟夫排列。例如， $(7, 3)$ -约瑟夫排列为 $(3, 6, 2, 7, 5, 1, 4)$ 。给定 N 和 K，编写程序求出 (N, K) -约瑟夫排列。

输入

第一行包含用空格分隔的 N 和 K。 $(1 \leq K \leq N \leq 5,000)$

输出

输出示例中所示的约瑟夫斯置换。

输入示例 1

```
7 3
```

示例输出 1

```
<3, 6, 2, 7, 5, 1, 4>
```

Silver 3

8 - Fibonacci function

Time limit	Memory limit
0.25 Sec	128 MB

Problem

The following source is a C++ function that calculates the Nth Fibonacci number.

```
int fibonacci(int n) {
    if (n == 0) {
        printf("0");
        return 0;
    } else if (n == 1) {
        printf("1");
        return 1;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}
```

Calling fibonacci(3) results in the following: fibonacci(3) calls fibonacci(2) and fibonacci(1) (the first call).

fibonacci(2) calls fibonacci(1) (the second call) and fibonacci(0).

The second call to fibonacci(1) outputs 1 and returns 1.

fibonacci(0) outputs 0 and returns 0. fibonacci(2) obtains the results of fibonacci(1) and fibonacci(0), and returns 1.

The first call to fibonacci(1) outputs 1 and returns 1.

fibonacci(3) obtains the results of fibonacci(2) and fibonacci(1), and returns 2.

1 is printed twice, and 0 is printed once. Given an integer N, write a program that determines how many times 0 and 1 are printed when fibonacci(N) is called.

Input

The first line contains the number of test cases, T. Each test case consists of a single line containing N. N is a natural number less than or equal to 40, or 0

Output

For each test case, output the number of times 0 is printed and the number of times 1 is printed, separated by a space

Enter Example 1

```
3 0 1  
3
```

Example Output 1

```
1 0 0 1  
1 2
```

Enter Example 2

```
2 6  
22
```

Example Output 2

```
5 8  
10946 17711
```

银级 3

8 - 斐波那契数列

时间限制

0.25 Sec

内存限制

128MB

问题

以下源代码是一个计算第 N 个斐波那契数的 C++ 函数

```
int fibonacci(int n) {
    if (n == 0) {
        printf("0");
        return 0;
    } else if (n == 1) {
        printf("1");
        return 1;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}
```

调用 fibonacci(3) 时，会发生如下过程：

fibonacci(3) 调用 fibonacci(2) 和 fibonacci(1) (第一次调用)。

fibonacci(2) 调用 fibonacci(1) (第二次调用) 和 fibonacci(0)。

第二次调用的 fibonacci(1) 输出 1 并返回 1。fibonacci(0) 输出 0 并返回 0。

fibonacci(2) 得到 fibonacci(1) 和 fibonacci(0) 的结果后，返回 1。

第一次调用的 fibonacci(1) 输出 1 并返回 1。

fibonacci(3) 得到 fibonacci(2)

和 fibonacci(1) 的结果后，返回 2

输入

第一行包含测试用例的数量 T。每个测试用例由一行组成，包含 N。N 是不超过 40 的自然数，或 0

输出

对于每个测试用例，输出 0 被打印的次数和 1 被打印的次数，用空格隔开

输入示例 1

```
3 0 1  
3
```

示例输出 1

```
1 0 0 1  
1 2
```

输入示例 2

```
2 6  
22
```

示例输出 2

```
5 8  
10946 17711
```

Silver 2

9 – Stock

Time limit	Memory limit
5 Sec	256 MB

Problem

Hong Joon is hooked on stocks these days. He has an exceptional eye for predicting the future, forecasting stock prices day by day, and he's always right. Every day, he does one of the following three things.

1. Buy one share
2. Sell as many shares as you want.
3. Do nothing.

Hong Joon has an exceptional eye for predicting the future, but he doesn't know how to maximize his own profit. Therefore, when he told you the stock price for each day, he asked you to calculate how much the maximum profit would be. For example, if the number of days is 3 and the daily stock prices are 10, 7, 6, the maximum profit becomes 0 because the price keeps decreasing. However, if the daily prices are 3, 5, 9, buying one share each of the first two days and selling all on the last day yields a profit of 10.

Input

The first line of input contains a natural number T , representing the number of test cases. For each test case, the first line contains a natural number N representing the number of days. On the second day, N natural numbers representing the stock price for each day are given in order, separated by spaces. The stock price for each day is less than or equal to 10,000. ($2 \leq N \leq 1,000,000$)

Output

For each test case, output one integer representing the maximum profit. The answer can be expressed as a signed 64-bit integer.

Enter Example 1

```
3
3
10 7 6
3
3 5 9
5
1 1 3 1 2
```

Example Output 1

0
10
5

银级 2

9 - 库存

时间限制 内存限制

5 Sec 256 MB

问题

最近洪俊迷上了炒股。他有着非凡的预判能力，能够准确预测未来的股价走势，而且每天的预测都不会出错。每天，他都会进行以下三种操作之一。

1. 购买一股
2. 随心所欲地包下任意数量的股份。
3. 无所作为

洪俊有着非凡的预判能力，但他并不知道怎样才能让自己的收益最大化。于是，当他把每天的股价都告诉你之后，就请你帮他计算一下能获得的最大利润是多少。例如，如果交易天数为 3 天，股价依次为 10、7、6，由于价格一直在下降，所以最大利润为 0。但是，如果股价依次为 3、5、9，第一天和第二天各买入一股，第三天全部卖出，就可以获得 10 的利润。

输入

第一行包含一个自然数 T ，表示测试用例的数量。

对于每个测试用例，第一行包含一个自然数 N ，表示天数。

第二行按顺序给出 N 个自然数，表示每天的股价，数之间用空格分隔。

每一天的股价都不超过 10,000。

输出

对于每个测试用例，输出一个整数，表示可以获得的最大利润。答案可用有符号 64 位整数表示。

输入示例 1

```
3
3
10 7 6
3
3 5 9
5
1 1 3 1 2
```

示例输出 1

0
10
5

Gold4

10 - Mutalisk

Time limit	Memory limit
2 Sec	512 MB

Problem

Subin is playing StarCraft with Kangho. Subin has one Mutalisk remaining, and Kangho has N SCVs remaining. Each SCV has its remaining health points, and they cannot attack the Mutalisk. Therefore, Subin has won the game. When the Mutalisk attacks, it can target three SCVs at once.

1. **The SCV attacked first loses 9 health points.**
2. The SCV attacked second loses 3 health points.
3. The SCV attacked third loses 1 health points.

If an SCV's health reaches 0 or below, it is destroyed immediately. You cannot attack the same SCV multiple times in a single attack. Given the remaining health of the SCVs, write a program to find the minimum number of attacks required to destroy all SCVs.

Input

The first line contains the number of SCVs, N. The second line contains the health of the N SCVs. Health is a natural number less than or equal to 60. ($1 \leq N \leq 3$)

Output

Output the minimum number of attacks required to destroy all SCVs on the first line.

Enter Example 1

```
3
12 10 4
```

Example Output 1

```
2
```

Enter Example 2

```
3
54 18 6
```

Example Output 2

6

Enter Example 3

1
60

Example Output 3

7

Enter Example 4

3
1 1 1

Example Output 4

1

Enter Example 5

2
60 40

Example Output 5

9

黄金 4

10 – Mutalisk

时间限制 内存限制

2Sec

512MB

问题

秀彬正在和康浩玩《星际争霸》。现在秀彬只剩下一只异虫飞龙 (Mutalisk)，而康浩还剩下 N 个 SCV。每个 SCV 都有一定的剩余生命值，并且它们无法攻击飞龙，因此秀彬已经赢下了这局游戏。当飞龙发动一次攻击时，可以同时攻击最多三个 SCV。

1. 第一个被攻击的 SCV 会失去 9 点生命
2. 第二个被攻击的 SCV 会失去 3 点生命值
3. 第三个被攻击的 SCV 会失去 1 点生命值

若 SCV 的生命值降至 0 或以下，则立即被摧毁。单次攻击中不可对同一 SCV 发动多次攻击。根据所有 SCV 的剩余生命值，编写程序找出摧毁所有 SCV 所需的最小攻击次数。

输入

第一行包含 SCV 的数量 N 。第二行包含 N 个 SCV 的生命值。生命值为不大于 60 的自然数。

输出

输出摧毁第一行所有 SCV 所需的最小攻击次数。

输入示例 1

```
3  
12 10 4
```

示例输出 1

```
2
```

输入示例 2

```
3  
54 18 6
```

示例输出 2

```
6
```

输入示例 3

```
1  
60
```

示例输出 3

```
7
```

输入示例 4

```
3  
1 1 1
```

示例输出 4

```
1
```

输入示例 5

```
2  
60 40
```

示例输出 5

```
9
```