2024 ACM Programming Contest

- Complete as many questions as you can during the contest.
- **Only one computer may be used per team.**
- No connection to the Internet may be made during the contest.
- All questions require you to read the test data from standard input and write results to standard output; you cannot use any non-contest files for input or output.
- The input to problems will consist of multiple test cases unless otherwise noted.
- Programming style is not considered in this contest. You are free to code in whatever style you prefer. Commenting code is not required.
- All communication with the judges will be handled by the PC2 environment.
- Allowed programming languages: C, C++, Java, and Python.
- **Java users: remember to call `nextLine()` on a scanner after reading in an integer when needed.**
- All programs will be re-compiled prior to testing with the judges' data.
- Output must match the sample format exactly. Do not print prompts unless asked to.
- Non-standard libraries cannot be used in your solutions. The Standard Template Library (STL) and string libraries are allowed. The standard Java API is available and can be used, except for those packages that are deemed dangerous by contest officials (e.g., that might generate a security violation)
- Hardcopy reference materials are allowed, but no electronic material. Calculators are fine, but no laptops, phones, USB drives, etc. are allowed.
- **Netbeans users: Remove package statement before you submit**

Judges decisions are to be considered final. No cheating will be tolerated.

Problem A - Mixed Fractions

You are part of a team developing software to help students learn basic mathematics. You are to write one part of that software, which is to display possibly improper fractions as mixed fractions. A proper fraction is one where the numerator is less than the denominator; a mixed fraction is a whole number followed by a proper fraction. For example the improper fraction 27/12 is equivalent to the mixed fraction 2 3/12. You should not reduce the fraction (i.e. don't change 3/12 to 1/4).

Input Input has one test case per line. Each test case contains two integers in the range $[1, 2^{31} - 1]$. The first number is the numerator and the second is the denominator. A line containing 0 0 will follow the last test case.

Output For each test case, display the resulting mixed fraction as a whole number followed by a proper fraction, using whitespace to separate the output tokens.

Sample Input

```
27 12
2460000 98400
3 4000
0 0
```

Sample Output

```
2 3 / 12
25 0 / 98400
0 3 / 4000
```

Problem B - Sensor Network

A wireless sensor network consists of autonomous sensors scattered in an environment where they monitor conditions such as temperature, sound, and pressure. Samantha is a researcher working on theAmazon Carbon-dioxide Measurement (ACM) project. In this project, a wireless sensor network in the Amazon rainforest gathers environmental information.

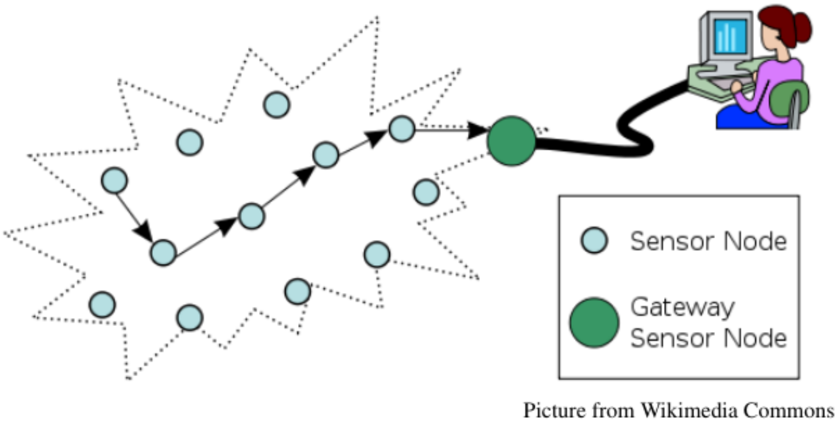

Figure 1: Sensor Network example pic

The Amazon rainforest stores an amount of carbon equivalent to a decade of global fossil fuel emissions, and it plays a crucial role in the world's oxygen-transfer processes. Because of the huge size of this forest, changes in the forest affect not only the local environment but also global climate by altering wind and ocean current patterns. The goal of the ACM project is to help scientists better understand earth's complex ecosystems and the impact of human activities. Samantha has an important hypothesis and to test her hypothesis, she needs to find a subset of sensors in which each pair of sensors can communicate directly with each other. A sensor can communicate directly with any other sensor having distance at most d from it. In order for her experiments to be as accurate as possible, Samantha wants to choose as many sensors as possible.

As one does not simply walk into the Amazon, Samantha cannot add new sensors or move those that are currently in place. So given the current locations of the sensors, she needs your help to find the largest subset satisfying her criteria. For simplicity, represent the location of each sensor as a point in a two-dimensional plane with the distance between two points being the usual Euclidean distance.

Input The input consists of a single test case. The first line contains two integers n and d where $1 \leq n \leq 100$ and $1 \leq d \leq 10000$, where n is the number of sensors available and d is the maximum distance between sensors that can communicate directly. Sensors are numbered 1 to n. Each of the next n lines contains two integers x and y where $-10000 \leq x, y \leq 10000$ indicating the sensor coordinates, starting with the first sensor.

Output Display a maximum subset of sensors in which each pair of sensors can communicate directly. The first line of output should be the size of the subset. The second line of output should be the (one- based) indices of the sensors in the subset. If there are multiple such subsets, any one of them will be accepted.

Sample Sensor Network Input 1

```
4 1
0 0
0 1
1 0
1 1
```

Sample Output 1

```
2
1 2
```

Sample Sensor Network Input 2

```
5 20
0 0
0 2
100 100
100 110
100 120
```

Sample Output 2

```
3
4 3 5
```

Problem C - Seven Segment Displays

In the modern computing world, seven segment displays are not generally used for new designs. However, in old designs, they appear everywhere. A marvel of modern engineering, it allowed systems to step away from nixie tubes and single LED indicators and output sensible and understandable data.

Seven segment displays contain seven "segments" and a decimal point. Segments are labeled `A-G` (including a decimal point as `DP`). `A` starts at the top most segment and follows in a clockwise diection, with G appearing as the middle segment. To display a specific number or letter, you can build a program (or even use an integrated circuit) to turn a combination of segments "on" and "off", which allows a number or letter to be displayed.

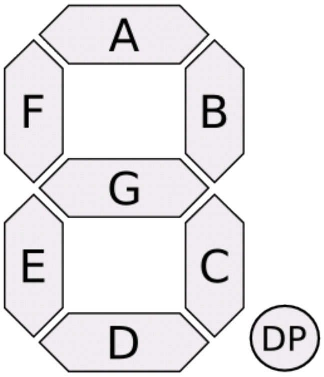

Figure 2: Seven Segment Display diagram

As a computer scientist, you're working with a group tasked with re-evaluating the code to control the automatic shutdown system of an RBMK nuclear reactor. After an incident occured near Pripyat, you've been tasked with creating a driver program to control numbers that appear on five seven segment displays. These displays will denote the current power output of the reactor.

Input Each input will be an integer $00000 \leq N \leq 99999$. The first digit (the digit further to the left) is considered the most significant digit. The digit further to the right is considered the least significant digit.

Output For each N, print the corresponding space-separated seven segment microcode. This should follow in the order of: `ABCDEFGDP`, with `DP` always being set as `0`. For each individual segment, denote if it's "on" (denoted by a `1`) or "off" (denoted by a `0`).

Sample Input

```
07734
```

Sample Output

```
11111100 11100000 11100000 11110010 01100110
```

Problem D - Chef and Party

Tonight, Chef would like to hold a party for his n friends. All friends are invited and they arrive at the party one by one in an arbitrary order. However, each friend has certain condition - they will not stay at a party with fewer than i people (excluding Chef). Assume an ideal order of arrival, when a friend arrives and there are strictly less than i other people (excluding Chef) at the party, this friend leaves the party; otherwise, this friend joins the party.

Help Chef estimate how successful the party can be — find the maximum number of his friends who could join the party (for an optimal choice of the order of arrivals).

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains a single integer n representing the number of friends Chef invites.
- The second line contains n space-separated integers $i_1, i_2, ..., i_n$ representing each friends conditional number of people at a party.

Output For each test case, print a single line containing one integer — the maximum number of Chef's friends who could join the party.

Example Input

```
3
2
0 0
6
3 1 0 0 5 5
3
1 2 3
```

Example Output

```
2
4
0
```

Problem E - 4 thought

Write a program which, given an integer n as input, will produce a mathematical expression whose solution is n. The solution is restricted to using exactly four 4's and exactly three of the binary operations selected from the set {*, +, -, /}. The number 4 is the ONLY number you can use. You are not allowed to concatenate fours to generate other numbers, such as 44 or 444.

For example given $n = 0$, a solution is $4 \times 4 - 4 \times 4 = 0$. Given $n = 7$, a solution is $4 + 4 - 4/4 = 7$. Division is considered truncating integer division, so that 1/4 is 0 (instead of 0.25). Assume the usual precedence of operations so that $4 + 4 \times 4 = 20$, not 32. Not all integer inputs have solutions using four 4's with the aforementioned restrictions (consider $n = 11$).

Hint: Using your forehead and some forethought should make an answer forthcoming. When in doubt use the fourth.

Input Input begins with an integer $1 \leq m \leq 1000$, indicating the number of test cases that follow. Each of the next m lines contain exactly one integer value for n in the range $-1000000 \leq n \leq 1000000$.

Output For each test case print one line of output containing either an equation using four 4's to reach the target number or the phrase no solution. Print the equation following the format of the sample output; use spaces to separate the numbers and symbols printed. If there is more than one such equation which evaluates to the target integer, print any one of them.

Sample 4 Thought input

```
5
9
0
7
11
24
```

Sample Output

```
4 + 4 + 4 / 4 = 9
4 * 4 - 4 * 4 = 0
4 + 4 - 4 / 4 = 7
no solution
4 * 4 + 4 + 4 = 24
```

Problem F - 8 Queens

In the game of chess, the queen is a powerful piece. It can attack by moving any number of spaces in its current row, in its column or diagonally.

In the eight queens puzzle, eight queens must be placed on a standard 8×8 chess board so that no queen can attack another. The center figure below shows an invalid solution; two queens can attack each other diagonally. The figure on the right shows a valid solution.

Given a description of a chess board, your job is to determine whether or not it represents a valid solution to the eight queens puzzle.

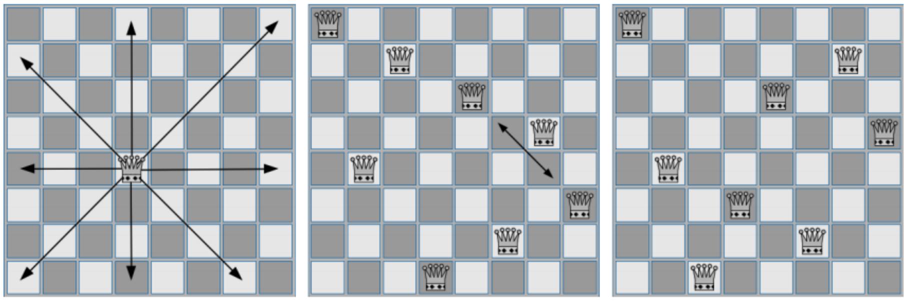

Figure 3: Queen movement (left), invalid solution (center), valid solution (right).

Input Input begins with an integer $1 \leq n \leq 20$ on a line by itself, indicating the number of test cases that follow. Each test case will contain a description of a single chess board, given as eight lines of eight characters each. Input lines will consist of only the characters ‘.’ and ‘*’. The ‘.’ character represents an empty space on the board, and the ‘*’ character represents a queen.

Output For each test case, print a single line of output. Print the word “valid” if the given chess board is a valid solution to the eight queens problem. Otherwise, print “invalid”.

Sample Input

```
2
*.......
..*.....
....*...
......*.
.*......
.......*
.....*..
...*....
*.......
......*.
....*...
.......*
.*......
...*....
.....*..
..*.....
```

Sample Output

```
invalid
valid
```

Problem G - Shotcube

Namco Bandai's video game 'Tales of Graces' introduces a puzzle minigame known as Shotcube. In this game, you have 9 cubes placed on a 7×7 square grid. The objective is to rearrange the cubes so that they all lie in a 3×3 square. In the video game, this 3×3 square must be in the exact center of the grid; for this problem, the 3×3 square may be anywhere on the grid.

The only way to rearrange the cubes is to "shoot" at them from the outside of the grid. If there is a cube adjacent to the edge of the grid, you may "shoot" it to push it in a straight line, in which it will keep moving until it hits another cube, at which point it stops. You may only shoot a cube if there is another cube in its path which stops it. If you shoot a cube in a direction such that one or more cubes are immediately behind it, all of those cubes will move in unison until the farthest cube hits another cube (after having travelled at least one grid square), at which point all of them stop.

Consider the grid below. The arrows mark the three legal shots; no other shots are legal, either because there is no cube adjacent to the edge of the grid, or because there is no cube present to stop the motion of the shot cubes. The three grids below that indicate the result of shooting the first row to the right, the first column down, and the fifth column down, respectively.

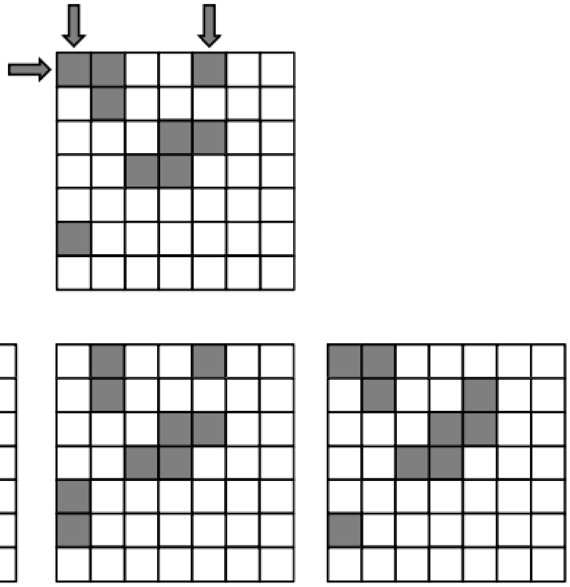

Figure 4: Illustration of legal shots and their outcomes.

Input Input begins with a line with a single integer T where $1 \leq T \leq 10000$ denoting the number of test cases. Each test case consists of 7 lines, each with 7 characters, representing the 7×7 grid. Each character is either a "." (period) or an "X". A "." indicates an empty square, while an "X" indicates a cube occupying the square. Each test case is guaranteed to have exactly 9 cubes. Each pair of test cases is separated by a single blank line.

Output For each test case, print out the minimum number of shots needed to arrange the cubes into a 3×3 square anywhere on the grid, following the rules described above. If it is impossible to do so, print out the number "-1" instead.

Sample Input

```
2
...X...
...X...
..X.X..
..XXX..
..X.X..
.......
.......

.......
....XXX
....XXX
......X
.......
.......
X....X.
```

Sample Output

```
-1
3
```

Problem H - Hitting Targets

A fundamental operation in computational geometry is determining whether two objects touch. For example, in a game that involves shooting, we want to determine if a player's shot hits a target. A shot is a two dimensional point, and a target is a two dimensional enclosed area. A shot hits a target if it is inside the target. The boundary of a target is inside the target. Since it is possible for targets to overlap, we want to identify how many targets a shot hits.

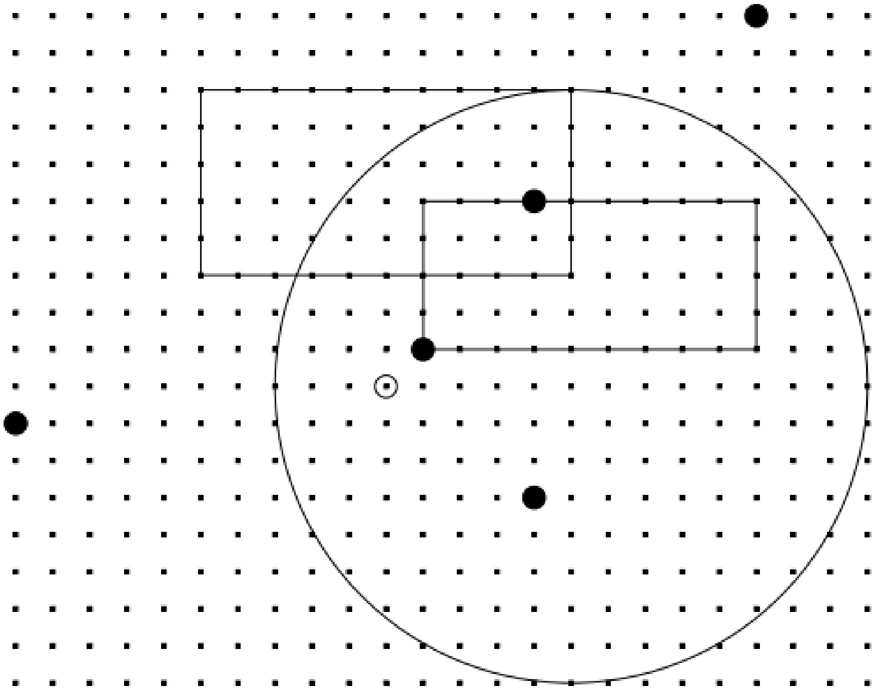

Figure 5: Illustration of Targets and shots

In the above illustration, the targets (large unfilled rectangles and circles) and shots (filled circles) show the the sample input. The origin (0, 0) is indicated by a small unfilled circle near the center.

Input Input starts with an integer $1 \leq m \leq 30$ indicating the number of targets. Each of the next m lines begins with the word rectangle or circle and then a description of the target boundary. A rectangular target's boundary is given as four integers x_1, y_1, x_2, y_2, where $x_1 < x_2$ and $y_1 < y_2$. The points (x_1, y_1) and (x_2, y_2) are the bottom-left and top-right corners of the rectangle, respectively. A circular target's boundary is given as three integers x y r. The center of the circle is at (x, y) and the $0 < r \leq 1000$ is the radius of the circle.

After the target descriptions is an integer $1 \leq n \leq 100$ indicating the number of shots that follow. The next n lines each contain two integers xy, indicating the coordinates of a shot. All x and y coordinates for targets and shots are in the range $[-1000, 1000]$.

Output For each of the n shots, print the total number of targets the shot hits.

Sample Input

```
3
rectangle 1 1 10 5
circle 5 0 8
rectangle -5 3 5 8
5
1 1
4 5
10 10
-10 -1
4 -3
```

Sample Output

```
2
3
0
0
1
```

Problem I - Morse Encode

Write a program that automatically converts English text to Morse code: Below is a table of the characters and their Morse Code representation.

A	.-
B	-...
C	-.-.
D	-..
E	.
F	..-.
G	--.
H	....
I	..
J	.---
K	-.-
L	.-..
M	--
N	-.
O	---
P	.--.
Q	--.-
R	.-.
S	...
T	-
U	..-
V	...-
W	.--
X	-..-
Y	-.--
Z	--..

Input

Input begins with an integer $1 \leq n \leq 20$ on a line by itself, indicating the number of test cases that follow. Each test case will be a string consisting of characters A - Z (upper case only, no spaces).

Output

For each test case, print a single line of output containing the morse code representation of the test case (placing a single space between each letter, see sample output).

Sample Input

```
1
HELLOWORLD
```

Sample Output

```
.... . .-.. .-.. - .- - .-. .-.. -..
```

Problem J - Das Blinkenlights

There are two lights that blink at regular intervals. When each one blinks, it turns on and then immediately back off; they don't toggle. They are both off at time $t = 0$. The first one blinks at $t = p, 2p, 3p, ...$ seconds; the second one blinks at $t = q, 2q, 3q, ...$ seconds. Once they start, they both keep blinking forever. It is very exciting to see them blink at the same time (on the same second). But your patience is going to run out eventually, in s seconds. Will they blink at same time between $t = 1$ and $t = s$ (inclusive)? Write a program that can answer this question, quick, before they start blinking again!

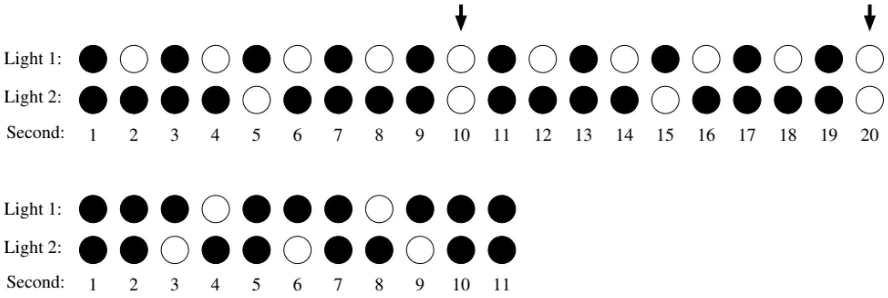

Figure 6: Illustration of the sample inputs. A black circle means the light is off, a white circle means the light blinks at that second. The arrows above point out times when both lights blink.

Input Input begins with an integer $1 \leq n \leq 20$ on a line by itself, indicating the number of test cases that follow. Each of the following n lines then consists of three space-separated integers p, q, s. The bounds are $1 \leq p,\ q \leq 100$ and $1 \leq s \leq 10000$. The first light blinks every p seconds, the second every q seconds. The value of s represents the maximum number of seconds to consider when determining if the two lights blink at the same time.

Output For each test case, output yes if the two lights blink on the same second between time 1 and time s, or no otherwise.

Sample Input

```
2
2 5 20
4 3 11
```

Sample Output

```
yes
no
```

Problem K - Poisonous Plants

There are a number of plants in a garden. Each of these plants has been treated with some amount of pesticide. After each day, if any plant has more pesticide than the plant on its left, being weaker than the left one, it dies.

You are given the initial values of the pesticide in each of the plants. Print the number of days after which no plant dies, i.e. the time after which there are no plants with more pesticide content than the plant to their left.

For example, pesticide levels $p = [3, 6, 2, 7, 5]$. Using a 1-indexed array, day 1 plants 2 and 4 die, leaving $p = [3, 2, 5]$. On day 2, plant 3 of the current array dies leaving $p = [3, 2]$. As there is no plant with a higher concentration of pesticide than the one to its left, plants stop dying after day 2.

Input The first line contains an integer which is the total number of test cases. The next line contains an integer n, and the size of the array p. The next line contains n space-separated integers $p[i]$ representing pesticide levels in each plant.

Output Output an integer equal to the number of days after which no plants die.

Constraints $1 \leq n \leq 10^5$

$0 \leq p[i] \leq 10^9$

Sample Input

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

Sample Output

```
2
```

Problem L - Recursive Digits

We define super digit of an integer x using the following rules:

- If x has only 1 digit, then its super digit is x.
- Otherwise, the super digit of x is equal to the super digit of the sum of the digits of x.

For example, the super digit of 9875 will be calculated as:

1. `super_digit(9875)` $\rightarrow 9 + 8 + 7 + 5 = 29$
2. `super_digit(29)` $\rightarrow 2 + 9 = 11$
3. `super_digit(11)` $\rightarrow 1 + 1 = 2$
4. `super_digit(2)` $\rightarrow= 2$

You are given two numbers n and k. The number p is created by concatenating the string n k times.

Continuing the above example where $n = 9875$, assume your value $k = 4$. Your initial $p = 9875\,9875\,9875\,9875$ (spaces added for clarity).

1. `superDigit(p)` $\rightarrow$ `superDigit(9875987598759875)` $\rightarrow 9+8+7+5+9+8+7+5+9+8+7+5+9+8+7+5 = 116$
2. `superDigit(116)` $\rightarrow 1 + 1 + 6 = 8$
3. `superDigit(p)` $= 8$

Input The first line contains the number of test cases. The next line contains two space separated integers, n and k.

Output Return the super digit of p, where p is created as described above.

Constraints $1 \leq n \leq 10^{100000}$

$1 \leq k \leq 10^5$

Sample Input

```
1
148 3
```

Sample Output

```
3
```