COMPUTER SCIENCES

George Fox University H.S. Programming Contest Division – I April 5-9, 2021 (practice problems)

General Notes

1. Do the problems in any order you like. They do not have to be done in order
(hint: the easiest problem may not be the first problem)

2. Scoring: While this is part of a week-long non-competitive practice set of problems... we will keep score consistent with the actual contest. The team who solves the most problems in the least amount of time with the least submissions wins. Each wrong submission will receive a 20 min time penalty that will only be added to the time score once the problem has been successfully solved. Time is calculated for each problem as the total time from the start of the contest to the time it was solved.

3. There is no extraneous input. All input is exactly as specified in the problem. Integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Do not welcome the user. Do not prompt for input. Follow the form exactly as given in the problem.
(hint: spaces? No spaces? What does spec say!)

5. All solutions must be a single source code file.

This page intentionally left blank

1. Runes

You are helping an archaeologist decipher some runes. He knows that this ancient society used a Base 10 system, and that they never start a number with a leading zero. He's figured out most of the digits as well as a few operators, but he needs your help to figure out the rest.

The professor will give you a simple math expression. He has converted all of the runes he knows into digits. The only operators he knows are addition (+), subtraction (-), and multiplication (*), so those are the only ones that will appear. Each number will be in the range from -999,999 to 999,999, and will consist of only the digits ‘0’-‘9’, possibly a leading ‘-’, and a few ‘?’s. The ‘?’s represent a digit rune that the professor doesn't know (never an operator, an ‘=’, or a leading ‘-’). All of the ‘?’s in an expression will represent the same digit (0-9), and it won't be one of the other given digits in the expression.

Given an expression, figure out the value of the rune represented by the question mark. If more than one digit works, give the lowest one. If no digit works, well, that's bad news for the professor, it means that he's got some of his runes wrong. Output -1 in that case.

Input: The first input will be an integer representing the number of cases T ($1 \leq T \leq 100$). Each case will consist of a single input string, of the form: `[number][op][number]=[number]`

Each [number] will consist of only the digits ‘0’-‘9’, with possibly a single leading minus ‘-’, and possibly some ‘?’s. No number will begin with a leading ‘0’ unless it is 0, there is no number that will begin with -0, and no number will have more than 6 characters (digits or ?s). The [op] will separate the first and second [number]s, and will be one of: +, - or *. The = will always be present between the second and third [number]s. There will be no spaces, tabs, or other characters. There is guaranteed to be at least one ? in every equation.

Output: Output the lowest digit that will make the equation work when substituted for the ?s, or output −1 if no digit will work. Output no extra spaces or blank lines.

Example Input
```
5
1+1=?
123*45?=5?088
-5?*-1=5?
19--45=5?
??*??=302?
```

Output to screen:
```
2
6
0
-1
5
```

This page intentionally left blank

2. Majority

The votes are in! Mathematicians world-wide have been polled, and each has chosen their favorite number between 1 and 1000. Your goal is to tally the votes and determine what the most popular number is.

If there is a tie for the greatest number of votes, choose the smallest number with that many votes.

Input: The first input will be an integer representing the number of cases, between 1 and 100 inclusive.

For each case, there will be a single line giving the number of votes V, 1 <= V <= 1000. Following that line will be V lines, each with a single integer vote between 1 and 1000.

Output: The most popular number as defined above.

Example Input

```
3
3
42
42
19
4
7
99
99
7
5
11
12
13
14
15
```

Output to screen:

```
42
7
11
```

This page intentionally left blank

3. Pushups

A friend of yours is on the cheer squad for their football team. Each time the team scores, the cheer squad does pushups, one for each point the team has scored so far. If the teams scores a touchdown (7 points), the squad does 7 pushups. If the team then scores a field goal (3 points), the cheer squad does 10 pushups. If the team then scores a safety (2 points), the squad will do 12 pushups. At the end of that game, the squad will end up having done 7+10+12=29 pushups!

You meet your friend after a game, and they say “Boy, am I tired! I did a total of n pushups at the game today!" and promptly collapse from exhaustion. Given n, the number of pushups, can you figure out how the team scored? More than one score may be possible. For example, for 29 pushups, the team could have scored 3, then 2, then 2, then 7, for a total of 14 points. If so, find the highest possible score.

Input

The input will start with a single integer giving the number of cases, between 1 and 20, inclusive. Each case will begin with two integers N and M, 1 <= N <= 5;000; 1 <= M <= 10, where N is the number of pushups the cheer squad did, and M is the number of ways a team can score points in that sport. The next input will be M unique integers S_i, 1 <= S_i <= 20, indicating the number of points the team gets for each kind of score. The scores are independent; a team can accrue scores in any order.

Output

For each test, output a single integer indicating the team's final score. If more than one final score can lead to the given number of pushups, output the largest one. If no final score can lead to the given number of pushups, then your friend must have miscounted. In this case, output ‘-1’. Output no extra spaces.

Example Input

```
4
29 3
7 3 2
15 1
1
16 1
1
6 2
3 1
```

Example Output to Screen

```
14
5
-1
3
```

This page intentionally left blank

4. Wormhole

With our time on Earth coming to an end, Cooper and Amelia have volunteered to undertake what could be the most important mission in human history: travelling beyond this galaxy to discover whether mankind has a future among the stars. Fortunately, astronomers have identified several potentially inhabitable planets and have also discovered that some of these planets have wormholes joining them, which effectively makes the travel distance between these wormhole connected planets zero. For all other planets, the travel distance between them is simply the Euclidean distance between the planets. Given the location of Earth, planets, and wormholes, find the shortest travel distance between any pairs of planets.

Input

- The first input will be a single integer, T (1 <= T <= 10) the number of cases.
- Each case consists of planets, wormholes, and a set of distance queries.
- The planets list for a case starts with a single integer, p (1 <= p <= 60), the number of planets. Following this are p input strings, where each string contains a planet name along with the planet's integer coordinates, i.e. name x y z (x, y, & z are all between 0 and $2*10^6$ inclusive) The names of the planets will consist only of ASCII letters and numbers, and will always start with an ASCII letter. Planet names are case-sensitive (Earth and earth are distinct planets). The length of a planet name will never be greater than 50 characters. All coordinates are given in parsecs.
- The wormholes list for a case starts with a single integer, w (0 <= w <= 40), the number of wormholes, followed by the list of w wormholes. Each wormhole consists of a string containing two planet names separated by a space. The first planet name marks the entrance of wormhole, and the second planet name marks the exit from the wormhole. The planets that mark wormholes will be chosen from the list of planets given in the preceding section. Note: you can't enter a wormhole at its exit.
- The queries list for a case starts with a single integer, q (1 <= q <= 20), the number of queries. Each query consists of a string containing two planet names separated by a space. Both planets will have been listed in the planet list.

Output

For each test case, output a line, “Case i:”, the number of the i^{th} test case. Then, for each query in that case, output a line that states “The distance from planet1 to planet2 is d parsecs.”, where the planets are the names from the query and d is the shortest possible travel distance between the two planets. Round d to the nearest integer.

Example Input

```
3
4
Earth 0 0 0
Proxima 5 0 0
Barnards 5 5 0
Sirius 0 5 0
2
Earth Barnards
Barnards Sirius
6
Earth Proxima
Earth Barnards
Earth Sirius
Proxima Earth
Barnards Earth
Sirius Earth
3
z1 0 0 0
z2 10 10 10
z3 10 0 0
1
z1 z2
3
z2 z1
z1 z2
z1 z3
2
Mars 12345 98765 87654
Jupiter 45678 65432 11111
0
1
Mars Jupiter
```

Example Output to Screen

```
Case 1:
The distance from Earth to Proxima is 5 parsecs.
The distance from Earth to Barnards is 0 parsecs.
The distance from Earth to Sirius is 0 parsecs.
The distance from Proxima to Earth is 5 parsecs.
The distance from Barnards to Earth is 5 parsecs.
The distance from Sirius to Earth is 5 parsecs.
Case 2:
The distance from z2 to z1 is 17 parsecs.
The distance from z1 to z2 is 0 parsecs.
The distance from z1 to z3 is 10 parsecs.
Case 3:
The distance from Mars to Jupiter is 89894 parsecs.
```

5. Number Game

Alice and Bob are playing a game on a line of N squares. The line is initially populated with one of each of the numbers from 1 to N. Alice and Bob take turns removing a single number from the line, subject to the restriction that a number may only be removed if it is not bordered by a higher number on either side. When the number is removed, the square that contained it is now empty. The winner is the player who removes the 1 from the line. Given an initial configuration, who will win, assuming Alice goes first and both of them play optimally?

Input

The first input will be a single integer T, 1 <= T <= 100, denoting the number of cases. Each case begins with a with a single integer N, 1 <= N <= 100, denoting the number of inputs to come. The last inputs for a case are the numbers from 1 to N, in a specified order for that case.

Output

For each test case, print the name of the winning player on a single line.

Example Input

```
4
4
2 1 3 4
4
1 3 2 4
3
1 3 2
6
2 5 1 6 4 3
```

Example Output to Screen

```
Bob
Alice
Bob
Alice
```

This page intentionally left blank

6. Wrench

Peter works at a factory. He is looking at a list of wrench sizes and needs to find the appropriately sized wrench for various screws and nuts and bolts to do his work. Normally, these sizes are specified using US Customary Unit notation such as 13/16", or 3/8", and so on. Another way to write 13/16" is 0.8125" But the reference sheets for various parts round the numbers in weird ways, and give approximations only, so for example 13/16" might turn into 0.812, or 0.813, or sometimes just 0.81, depending on the method of rounding. Given that Peter is looking for a wrench of size A/B", and it is customary for B to be a power of 2, help Peter find the correct wrench size, where A is a positive integer and B is the minimum possible base (power of 2).

Input

The first input is an integer that represents the number of cases, T, with 1 <= T <= 100. Each case is a single decimal number representing a wrench size, with at most six digits after the decimal point. There need not always be a decimal point. The input value will be greater than zero.

Output

A/B", or C A/B", or C", where B is the minimal power of two such that the exact decimal representation rounded to the number of decimal digits of the input matches the input, using one of the following rounding rules: round up (ceiling), round down (or truncate), or round-to-nearest. The wrench will be less than or equal to 10 inches. There will always be a valid power of two less than or equal to 128.

Example Input

```
10
0.81
.8125
0.37
2
2.4
2.99
2.40
1.27
4.
9.242187
```

Example Output to Screen

```
13/16"
13/16"
3/8"
2"
2 3/8"
2 63/64"
2 13/32"
1 17/64"
4"
9 31/128"
```

This page intentionally left blank