

Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

A. Crashing Robots

In a modernized warehouse, robots are used to fetch the goods. Careful planning is needed to ensure that the robots reach their destinations without crashing into each other. Of course, all warehouses are rectangular, and all robots occupy a circular floor space with a diameter of 1 meter. Assume there are N robots, numbered from 1 through N . You will get to know the position and orientation of each robot, and all the instructions, which are carefully (and mindlessly) followed by the robots. Instructions are processed in the order they come. No two robots move simultaneously; a robot always completes its move before the next one starts moving.

A robot crashes with a wall if it attempts to move outside the area of the warehouse, and two robots crash with each other if they ever try to occupy the same spot.

The first line of **input** is K , the number of test cases. Each test case starts with one line consisting of two integers, $1 \leq A, B \leq 100$, giving the size of the warehouse in meters. A is the length in the EW-direction, and B in the NS-direction. The second line contains two integers, $1 \leq N, M \leq 100$, denoting the numbers of robots and instructions respectively. Then follow N lines with two integers, $1 \leq X_i \leq A$, $1 \leq Y_i \leq B$ and one letter (N, S, E or W), giving the starting position and direction of each robot, in order from 1 through N . No two robots start at the same position.

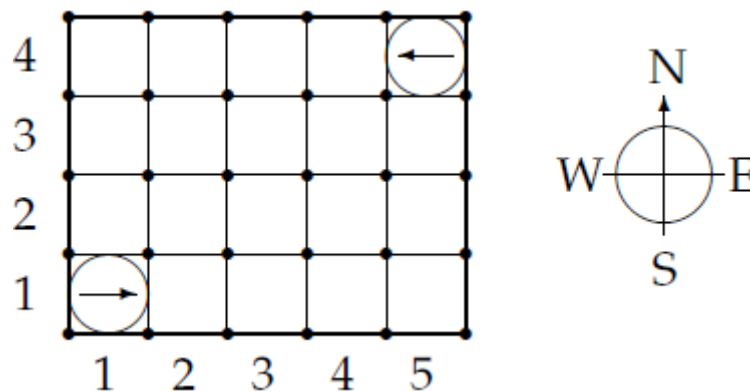


Figure 1: The starting positions of the robots in the sample warehouse

Finally there are M lines, giving the instructions in sequential order.

An instruction has the following format:

< robot #> < action> < repeat>

Where <action> is one of

- L: turn left 90 degrees,
- R: turn right 90 degrees, or
- F: move forward one meter,



and $1 \leq \langle \text{repeat} \rangle \leq 100$ is the number of times the robot should perform this single move.

Output one line for each test case:

- Robot i crashes into the wall, if robot i crashes into a wall. (A robot crashes into a wall if $X_i = 0$, $X_i = A + 1$, $Y_i = 0$ or $Y_i = B + 1$.)
- Robot i crashes into robot j , if robots i and j crash, and i is the moving robot.
- OK, if no crashing occurs.

Only the first crash is to be reported.

Sample Input

```

4
5 4
2 2
1 1 E
5 4 W
1 F 7
2 F 7
5 4
2 4
1 1 E
5 4 W
1 F 3
2 F 1
1 L 1
1 F 3
5 4
2 2
1 1 E
5 4 W
1 L 96
1 F 2
5 4
2 3
1 1 E
5 4 W
1 F 4
1 L 1
1 F 20

```

Sample Output

```

Robot 1 crashes into the wall
Robot 1 crashes into robot 2
OK
Robot 1 crashes into robot 2

```



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

B. The Embarrassed Cryptographer

The young and very promising cryptographer Odd Even has implemented the security module of a large system with thousands of users, which is now in use in his company. The cryptographic keys are created from the product of two primes, and are believed to be secure because there is no known method for factoring such a product effectively. What Odd Even did not think of, was that both factors in a key should be large, not just their product. It is now possible that some of the users of the system have weak keys. In a desperate attempt not to be fired, Odd Even secretly goes through all the users keys, to check if they are strong enough. He uses his very powerful Atari, and is especially careful when checking his boss' key.

$$\begin{aligned}
 N &= p \cdot q \quad \text{☹} \\
 f &= (p-1)(q-1) \\
 e < f, \quad \gcd(e, f) &= 1 \\
 d < f, \quad d \cdot e &\equiv 1 \pmod{f}
 \end{aligned}$$

Input specifications

The input consists of no more than 20 test cases. Each test case is a line with the integers $4 \leq K \leq 10^{100}$ and $2 \leq L \leq 10^6$. K is the key itself, a product of two primes. L is the wanted minimum size of the factors in the key. The input set is terminated by a case where $K = 0$ and $L = 0$.

Output specifications

For each number K , if one of its factors are strictly less than the required L , your program should output "BAD p ", where p is the smallest factor in K . Otherwise, it should output "GOOD". Cases should be separated by a line-break.

Sample input	Output for sample input
143 10	GOOD
143 20	BAD 11
667 20	GOOD
667 30	BAD 23
2573 30	GOOD
2573 40	BAD 31
0 0	



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

C. Electrical Outlets



Roy has just moved into a new apartment. Well, actually the apartment itself is not very new, even dating back to the days before people had electricity in their houses. Because of this, Roy's apartment has only one single wall outlet, so Roy can only power one of his electrical appliances at a time.

Roy likes to watch TV as he works on his computer, and to listen to his HiFi system (on high volume) while he vacuums, so using just the single outlet is not an option. Actually, he wants to have all his appliances connected to a powered outlet, all the time. The answer, of course, is power strips, and Roy has some old ones that he used in his old apartment. However, that apartment had many more wall outlets, so he is not sure whether his power strips will provide him with enough outlets now.

Your task is to help Roy compute how many appliances he can provide with electricity, given a set of power strips. Note that without any power strips, Roy can power one single appliance through the wall outlet. Also, remember that a power strip has to be powered itself to be of any use.

Input will start with a single integer $1 \leq N \leq 20$, indicating the number of test cases to follow. Then follow N lines, each describing a test case. Each test case starts with an integer $1 \leq K \leq 10$, indicating the number of power strips in the test case. Then follow, on the same line, K integers separated by single spaces, $O_1 O_2 \dots O_K$, where $2 \leq O_i \leq 10$, indicating the number of outlets in each power strip.

Output one line per test case, with the maximum number of appliances that can be powered.

Sample input	Sample output
3	7
3 2 3 4	31
10 4 4 4 4 4 4 4 4 4	37
4 10 10 10 10	

Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

D. Playground

George has $K \leq 20$ steel wires shaped in the form of half-circles, with radii a_1, a_2, \dots, a_K . They can be soldered (connected) at the ends, in any angle. Is it possible for George to make a closed shape out of these wires? He does not have to use all the wires. The wires can be combined at any angle, but may not intersect. Beware of floating point errors.



Input specifications

Each data set consists of a number $0 < K \leq 20$ on a line by itself, followed by a line of K space-separated numbers a_i . Each number is in the range $0 < a_i < 10^7$, and has at most 3 digits after the decimal point. The input will be terminated by a zero on a line by itself.

Output specifications

For each test case, there should be one word on a line by itself; “YES” if it is possible to make a simple connected figure out of the given arcs, and “NO” if it isn’t.

Sample input	Sample output
1	NO
4.000	YES
2	NO
1.000 1.000	YES
3	
1.455 2.958 4.424	
7	
1.230 2.577 3.411 2.968 5.301 4.398 6.777	
0	



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

E. Conformity

Frosh commencing their studies at Waterloo have diverse interests, as evidenced by their desire to take various combinations of courses from among those available.

University administrators are uncomfortable with this situation, and therefore wish to offer a conformity prize to frosh who choose one of the most popular combinations of courses. How many frosh will win the prize?

The input consists of several test cases followed by a line containing 0. Each test case begins with an integer $1 \leq n \leq 10000$, the number of frosh. For each frosh, a line follows containing the course numbers of five distinct courses selected by the frosh. Each course number is an integer between 100 and 499.

The popularity of a combination is the number of frosh selecting exactly the same combination of courses. A combination of courses is considered most popular if no other combination has higher popularity. For each line of input, you should output a single line giving the total number of students taking some combination of courses that is most popular.

Sample Input

```
3
100 101 102 103 488
100 200 300 101 102
103 102 101 488 100
3
200 202 204 206 208
123 234 345 456 321
100 200 300 400 444
0
```

Output for Sample Input

```
2
3
```

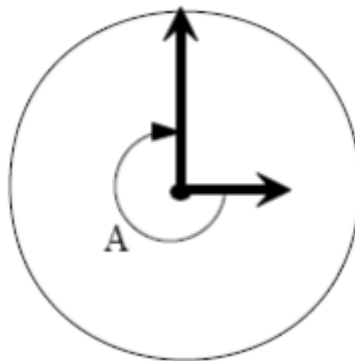


Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

F. Context-Free Clock

You recently installed a stylish clock in your office that is perfectly round and has no markings that identify its orientation. After accidentally bumping it, you realized that 12 o'clock might no longer be at the top. Nonetheless, you want to figure out what time it is. Fortunately you recently overheard a coworker giving the time and you have a protractor and can measure the angle between the hour and minute hands.

Your program should print the first time that has the correct angle between the hour and minute hands and that is on or after the overheard time. The angle (0 to 359 degrees, inclusive) will be measured clockwise from the hour hand to the minute hand. Assume that the clock hands move smoothly.



Input will consist of one test case per line, of the form A HH:MM:SS where A is the integral number of degrees that must be traversed clockwise to get from the hour hand to the minute hand and HH : MM : SS is the overheard time in 24 hour form. $0 \leq A \leq 359$, $0 \leq HH \leq 23$, $0 \leq MM \leq 59$, and $0 \leq SS \leq 59$. HH, MM, and SS will be exactly two digits with a leading zero if necessary. End of input will be signaled by the line -1 00:00:00

Output will consist of one line per test case, of the form HH:MM:SS where HH : MM : SS is the first time on or after the input time where the angle from the hour hand to the minute hand is exactly A degrees, rounded down to the nearest second. HH, MM, and SS should be zero padded to two digits and in the same range as the input (0 . . . 23, 0 . . . 59, and 0 . . . 59 respectively).

Input	Output
270 14:45:00	15:00:00
0 12:00:00	12:00:00
0 12:00:01	13:05:27
300 13:30:00	14:00:00
180 08:30:00	09:16:21
-1 00:00:00	



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 5 октомври 2013 г.

G. Necklace Decomposition

The set of cyclic rotations of a string are the strings obtained by embedding the string clockwise on a ring, with the first character following on the last, starting at any character position and moving clockwise on the ring until the character preceding the starting character is reached. A string is a necklace if it is the lexicographically smallest among all its cyclic rotations. For instance, for the string 01011 the cyclic rotations are (10110,01101,11010,10101,01011), and furthermore 01011 is the smallest string and hence, a necklace.

Any string S can be written in a unique way as a concatenation $S = T_1 T_2 \dots T_k$ of necklaces T_i such that $T_{i+1} < T_i$ for all $i = 1, \dots, k - 1$, and $T_i T_{i+1}$ is not a necklace for any $i = 1, \dots, k - 1$. This representation is called the necklace decomposition of the string S , and your task is to find it.

The relation $<$ on two strings is the lexicographical order and has the usual interpretation: $A < B$ if A is a proper prefix of B or if A is equal to B in the first $j - 1$ positions but smaller in the j th position for some j . For instance, $001 < 0010$ and $1101011 < 1101100$.

Input

On the first line of the input is a single positive integer n , telling the number of test scenarios to follow. Each scenario consists of one line containing a non-empty string of zeros and ones of length at most 100.

Output

For each scenario, output one line containing the necklace decomposition of the string. The necklaces should be written as '(necklace)'.

Sample Input:

```
5
0
0101
0001
0010
11101111011
```

Sample Output:

```
(0)
(0101)
(0001)
(001) (0)
(111) (01111) (011)
```