Mock CCC 2014 by Alex and Timothy Li (Junior Division)

These problems are available for online grading on http://wcipeg.com/contests, however, due to latency issues with the server, they have been released by PDF as an alternative. If you would like your solutions to be graded before the CCC (February 25, 2014), either wait for our online judge to respond, or feel free to email your solutions to wcipeg@gmail.com (attach your source files to the same email message, naming it something clear like "j1.cpp", "j2.java", etc.). We'll be glad to grade it for you ASAP and reply with your score and grading details before the real contest. Analyses, test data, and solutions will be released Sunday night (February 23rd, 2014).

In this contest, there are 5 problems of increasing difficulty, each scored out of 15, for a total of 75 points. You should give yourself 3 hours to complete the problems. All I/O are from standard input and output. Do not read from any files. Remember that on the real CCC, machine-readable materials (e.g. forums, programs you have written previously.) are not allowed, but language reference (such as www.cplusplus.com or http://docs.oracle.com/javase/7/docs/api/), as well as books and printed-materials are allowed.

We've put a lot of work into these problems, and we hope you enjoy them.

Good luck!

Problem J1: Love Overdose

Alice has a crush on her friend Bob, but she is too shy to express it. From an ancient recipe passed down through her family, she has concocted a vial of love potion. She plans to slip this potion into Bob's food every day so he will love her forever. The only problem is that the potion is so potent, taking too much can make Bob overdose, resulting in him being utterly indifferent towards her.

According to a note on the recipe, if some amount of the potion is ingested one day, only 50% percent of that amount is retained in the body on the next day, and 0% of that amount is retained on the day after that. You are given A and B, the amount of love potion in mL that Alice gives to Bob on the first and second day, respectively, and R, the recommended dosage in mL, according to her recipe. Write a program to determine whether Bob overdoses on the first day, the second day, or neither. Note that if Bob overdoses on the first day, he cannot overdose again on the second.

Input Format

Three lines, containing the three non-negative integers A, B, and R, each no greater than 1000.

Output Format

If he overdoses on the first day, output "`Bob overdoses on day 1.`" If he overdoses on the second day, output "`Bob overdoses on day 2.`" Otherwise, output "`Bob never overdoses.`"

Sample Input 1

```
100
200
249
```

Sample Output 1

```
Bob overdoses on day 2.
```

Sample Input 2

```
50
100
125
```

Sample Output 2

```
Bob never overdoses.
```

Explanation

For the first example, Alice slips Bob 100 mL of the potion on day one, of which, only 50 mL remain in Bob's body on day two. 50 mL is added on top of the 200 mL that he ingests on day two to make 250 mL, which is barely enough to make him overdose the recommended dosage of 249 mL.

For the second example, the amount from day one reduces to 25 mL on day two, when he is given another 100 mL to yield a total of 125 mL for the second day. 125 mL happens to be just the right dosage, so he does not overdose.

Problem J2: Rope Unfolding

Alice's love potion failed because Bob overdosed. Now, her new tactic is to impress Bob. Bob is a computer scientist who specializes in ropes. In computer science, a rope is a data structure used for efficiently manipulating strings. Bob has a rope of characters. However, it's too long to fit in his pocket, so he folds it into N segments. Each fold is at a letter which is not the last. The fold takes the rest of the rope after that letter, reverses it, and puts it above the current letter. For example, the rope "`ABRACADABRA`", when folded "`AB|RACA|DAB|RA`", yields:

```
 AR
DAB
ACAR
  AB
```

Alice notices that Bob often gets a headache pulling the rope out of his pocket and spending a long time untangling it. She concludes that if she finds a way to unfold it for Bob, Bob will be so impressed that he can't help but fall madly in love with her. You must write a program to help Alice!

Input Format

The first line of input will contain the positive integer N ($1 \leq N \leq 100$), the number of lines to follow. The next N lines each contain a segment of the folded rope.

The entire, unfolded rope will be no greater than 100 characters, and will only consist of uppercase letters from 'A' to 'Z'.

Output Format

The output should contain one line — the unfolded rope.

Sample Input 1

```
4
 AR
DAB
ACAR
  AB
```

Sample Output 1

```
ABRACADABRA
```

Sample Input 2

```
5
D
LRO
  W
  OL
 HEL
```

Sample Output 2

```
HELLOWORLD
```

Problem J3: Clock Hands

Alice had just dug up an old time capsule that she had buried in her backyard many years ago. Inside, she discovered an exquisite clock from when she was a young girl. When she buried the clock, she had removed the batteries so that the hands would remain in that position forever. This was done so she could remind her future self of the exact time at which she met her eternal love interest — Bob. Unfortunately, all but the 12 o'clock marking had worn away over the years. With a protractor handy, Alice is determined to find out the exact time that Bob came into her life.

Measuring the hour hand from the centre of the clock, Alice determined that it was H degrees clockwise from the 12 o'clock marking. Alice then measured the angle of the minute hand, determining that it was M degrees clockwise *relative to the hour hand*. Finally, she measured the second hand, determining that it was S degrees clockwise *relative to the minute hand*. The three angle measurements will all be non-negative real numbers greater than or equal to 0, but less than 360. You may assume that the angles will yield valid times.

Input Format

Three lines, containing the three values H, M, and S.

Output Format

Output the time, in the format `HH:MM:SS`, where HH ($1 \leq HH \leq 12$) is the number of hours, MM ($0 \leq MM < 60$) is the number of minutes, and SS ($0 \leq SS < 60$) is the number of seconds. These values each must be padded to two-digits with leading zeros. Round the time to the nearest second.

Sample Input 1

```
195
345
180
```

Sample Output 1

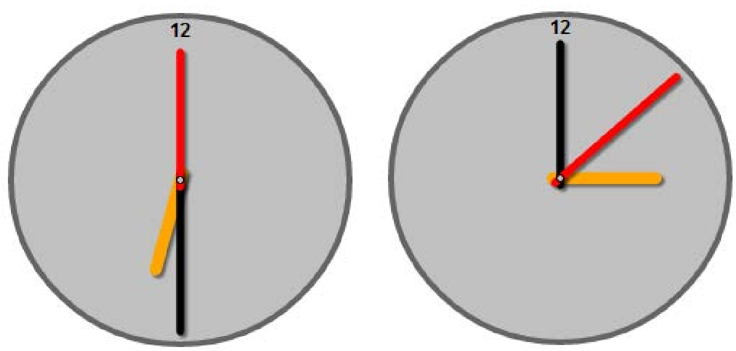
```
06:30:00
```

Sample Input 2

```
90.066667
270.733333
47.200000
```

Sample Output 2

```
03:00:08
```



Explanation

The above diagrams depict Alice's clock for the sample inputs. Orange is hour hand, black is the minute hand, and red is the second hand.

Problem J4: Selecting Shifts

Alice figured out where Bob works, and has obtained a job there so she can secretly admire him at work. Because she is new, the position is only part-time, and she is always called in to shifts by her boss. Whenever this happens, she is offered various shifts to pick from, starting and ending at different times of the day. Alice knows that Bob works from *T1* to *T2* on days that she's called in.

Alice's employer has given her a list of the shifts that she can choose from in order of decreasing pay. Given this list, Alice would like to know which one would result in the longest period of time that both of them are simultaneously on the job. If multiple shifts yield the same time spent with Bob, she would like to know the highest paying out of those (i.e. the one that appears earliest in the list).

Input Format

Line 1 contains *T1* and *T2*, the time that Bob starts and ends his shift.
Line 2 contains the integer N ($1 \leq N \leq 100$), the number of shifts that Alice is being offered.
The next N lines each contain two times — the start and end times of that shift. The shifts are ordered by decreasing pay.
Times in the input will be in the 7 character long format as depicted in the sample input below, and will all describe times within the same 24 hour day (between 12:00AM and 11:59PM, inclusive). Every shift described will have the start time strictly before the end time.

Output Format

The start and end time of the shift that Alice should pick to maximize the time she gets to spend on the job with Bob. If there are multiple answers, pick the one that is higher up on the list. If none of the shifts overlap with Bob shift, output "`Call in a sick day.`" (without quotes), since Alice has no interest in wasting her time at work when Bob is not there.

Sample Input

```
10:30AM 07:00PM
4
05:30AM 10:00AM
11:00AM 08:15PM
01:45PM 05:00PM
08:00PM 11:00PM
```

Sample Output

```
11:00AM 8:15PM
```

Explanation

Bob works from 10:30AM to 7:00PM. If Alice picks the shift from 1:45PM to 5:00PM, she will spend 3 hours and 15 minutes on the job with Bob. If Alice picks the shift from 11:00AM to 8:15PM, she will spend 8 hours on the job with Bob (from 11:00AM to 7:00PM, when Bob gets off). The shift from 5:30AM to 10:00AM ends before Bob even goes in to work, and the shift at 8:00PM starts after Bob has already left.

Problem J5: Spacetime Surfer

Alice's love for Bob is everlasting. Unfortunately, they are fated to be apart in the current time. Alas, her burning love cannot be stopped by such a small obstacle. To reach Bob, Alice has acquired a time machine with T ($1 \leq T \leq 10$) settings, along with maps of the terrain from each of those time periods. One of those periods is the present day. Alice and Bob live in a rectangular world that's R units tall by C units wide ($1 \leq R, C \leq 100$). In their world, places where they may walk are denoted by '`.`'s. and obstacles (places where they may not walk) are denoted by '`X`'s.

From Alice's perspective of time, it takes 1 second to move in any of the four directions adjacent to her (up, down, left, or right). It also takes her 1 second to travel to any one of the T time periods supported by her time machine. Whenever she travels through time, she always lands exactly on the same location. Of course, she cannot travel to another time if her location in that time is occupied by an obstacle. Alice would like to know the shortest time (from her own perspective) that it takes to get from her location in present day to Bob's location in present day.

Input Format

The first line of input contains the integers R, C, and T, the dimensions of the map and the number of time periods accessible by Alice's time machine.

The following contains T maps, each R rows by C columns. The first of these maps describes the present day. In the present day map, '`A`' indicates Alice's current position and '`B`' indicates Bob's current position. It is guaranteed that Alice cannot reach Bob from traversing only the present day setting.

Output Format

The shortest time that it takes for Alice to reach Bob, in seconds from Alice's perspective. If this is not possible, output -1.

Sample Input 1

```
3 3 2
AXX
.X.
XXB
XXX
...
XXX
```

Sample Output 1

```
6
```

Explanation

There are two time settings, depicted below:

```
(Present Day)
  Setting 0:    Setting 1:
     AXX           XXX
     .X.           ...
     XXB           XXX
```

The best path Alice can take is: down → travel through time → right → right → travel back to the present day → down to reach Bob.

Sample Input 2

```
2 5 3
BXXA.
XXX.X
.XXXX
..XXX
X...X
X.X.X
```

Sample Output 2

```
8
```

Explanation

The three time settings are:

```
(Present Day)
  Setting 0:    Setting 1:     Setting 2:
    BXXA.         .XXXX          X...X
    XXX.X         ..XXX          X.X.X
```

The best path she can take is: travel to setting 2 → left → left → down → travel to setting 1 → left → up → travel to present day, on top of Bob.