6562 Bowling Score Assistant

Bowling is a very popular sport in the U.S.. But projecting frames needed for winning a game is a bit painful for non-ACMers, so you are tasked with writing a program to compute this for them.

For anyone who has not bowled or has forgotten how the score is computed, there is a description of how to total a bowling score on the next couple pages, following the sample input and output. The main idea is that in order to win, you must get a higher score than your competitor.

Given your opponents final score and the number of pins you knocked down on each ball through the eighth frame, you are to compute what you need to win (if it is even possible). If you cant possibly win, print "impossible", otherwise print the sequence of rolls that will allow you to win that is first in lexicographical order. That is, the one with the lowest first roll, lowest second roll given the first roll, lowest third roll given the first two, etc..

Input

Input for this problem will be a series of test cases. Each case will consist of your opponents final score followed by the number of pins knocked down with each roll of the ball for your first eight frames. End-of-input will be indicated by a negative number. All other values on the file will be valid game pin counts. Your opponents score will be an integer from 0 to 300, and the numbers of pins knocked down on your rolls will all be nonnegative integers less than or equal to 10 (the 10 being the number of pins standing at the start of each frame). The sample input below is neatly formatted only to make it clear how it matches the examples that follow. You will need to process the input carefully in order to determine where one test case ends and the next begins, because you can not assume any format in terms of spacing and line breaks in the actual input file!

Output

For each input case, print the rolls needed to win the game. Follow this format exactly: "Case", one space, the case number, a colon and one space, and the answer for that case given as either "impossible" or the required numbers of pins, with exactly one space between each pair of numbers and no trailing spaces.

How to keep score in bowling:

There are ten pins that you attempt to knock down by rolling a ball at them.

For each set of ten pins, if you do not knock them all down with your first ball, you get a second try.

The one or two balls you roll at the set of ten pins is called a "frame," and there are ten frames in a game.

If you knock all the pins down with your first roll, that is called a "strike." If you knock all the pins down, but it takes both rolls, that is called a "spare."

The score is computed as the sum of the pins you knock down, but there are bonuses for spares and strikes which are computed as described below. If you never get a spare or a strike, your score is simply the total of your twenty rolls.

If you get a strike, you double count the score of the next two rolls. If you get a spare, you double count the next one roll.

The tenth frame is special in that the "next" roll(s) would not normally exist for a strike or a spare in the tenth frame. If you get a spare in the tenth frame, you get one more roll at a fresh set of ten pins which is simply added to the 10 for your spare. If you get a strike in the tenth frame, you get two

more rolls. If the first of these is also a strike, you get your second roll at another fresh set of ten pins. Note, however, that you dont go forever if you keep getting strikes. You only get the two rolls after the first strike in the tenth frame, and they are simply added to the 10 for the first strike, making a maximum score of 30 for the tenth frame (and for any other frame).

Thus, you can have three rolls in the tenth frame, so the maximum number of rolls in a game is $2 * 9 + 3 = 21$. The minimum number of rolls occurs if you get a strike for each of the first nine frames, but then do not get a strike or a spare in the tenth. This results in $9 + 2 = 11$ rolls.

Examples: We give the number of pins knocked down for each roll for ten frames and then compute the score.

Example 1:

1	2	3	4	5	6	7	8	9	10
4 3	6 2	4 2	8 1	4 5	6 2	7 2	9 0	0 5	6 3

Since there are no spares or strikes, the score is simply the sum of all these numbers which is 79.

Example 2:

1	2	3	4	5	6	7	8	9	10
4 6	6 2	4 2	8 1	10	6 2	7 2	9 0	0 5	6 3

This is similar to example 1, but there is a spare in frame 1 and a strike in frame 5. Note that we have a total of 10 in frame 1 before double counting the 6 (first roll) in frame 3. Also, because of the strike in frame 5, we double count the two rolls in frame 6. The final score is 97.

Example 3:

1	2	3	4	5	6	7	8	9	10
4 6	6 2	4 2	8 1	10	10	7 2	9 0	0 5	6 3

This is similar to example 2, but there is a strike in frame 6. Note that we have added two more pins being knocked down, so the base score is 85, instead of 83 as in example 2. As before, we double count the first roll in frame 2, which gives us 91. Next, we double count the two rolls after the strike in frame 5 which are 10 and 7. This gives us 108. Finally we double count the two rolls in frame 7, which gives us 117.

Example 4:

1	2	3	4	5	6	7	8	9	10
4 6	6 2	4 2	8 1	10	10	7 2	9 0	0 5	6 4 7

This is similar to example 3, but there is a spare in frame 10. The extra pin on the second roll in the tenth frame would boost the score to 118 (same computations as in example 3), but we get to add the 7 on the last roll to the spare, so the total is 125.

Example 5 (maximum score, a "perfect game"):

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

The total score for each frame is 30 since we get the 10 for the strike and the next two rolls which are both 10. This gives us the maximum score of 300.

Sample Input

```
80
4 3    6 2   4 2
8 1    4 5   6 2
7 2    9 0

215    6 2   4 2
4 6    10    6 2
8 1    9 0
7 2

299
10 10 10 10
10 10 10 10

-1
```

Sample Output

```
Case 1: 0 0 0 10 6
Case 2: impossible
Case 3: 10 10 10 10
```