4893 Dice Game

A certain game involves rolling five dice and scoring according to a score sheet. At each turn, the player rolls all five dice. Next, the player considers the roll and potentially removes and re-rolls some or all of the dice with the intention of potentially improving the score. This happens again, and after two re-rolls the final dice are used for scoring.

The score sheet is divided into two halves, the top half and (wait for it!) the bottom half. The bottom half is organized like various poker hands, such as three-of-a-kind and a full-house. The problem here only concerns the top half of the score sheet.

In the top half of the sheet, the player takes the final roll and adds up the dice according to whether they want "ones", "twos", and so on. In other words, if the five dice show 3-4-3-5-2, the player could opt to fill in the "threes" slot, and then adds the two occurrences of threes that are showing, for a total score of six. As another example, if a player were to roll 2-2-3-2-6, the score could be recorded as six in the twos slot. The player must fill in each of the slots: ones, twos, . . . up to the sixes slot. When all of the slots are filled, the total for the upper half is the sum of the scores in each slot. If a player scores a total of at least 63 points in the upper half, a bonus of 35 points is added to the upper section score. This 63 points corresponds to three-of-a-kind for each of the six dice faces. A common way to get the bonus is rolling four or five of a larger number so that fewer of the smaller numbers are needed (a player can earn the bonus even if he or she scores a "0" in an upper section slot).

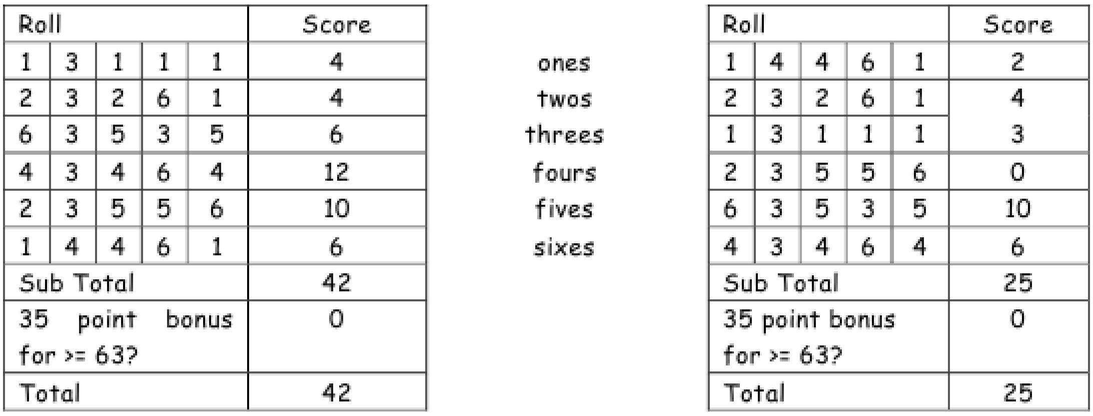
Since there are six upper half slots to fill, it will require six rolls of the dice. When actually playing the game, one of the variables is when to fill in each slot, and once an entry is placed on the score sheet it remains for the remainder of the game. So a player rolling 6-6-3-6-2 should probably score this as an 18 in the six slot, not a three in the three slot or a two in the two slot. In our case we have advance knowledge of all dice rolls, so we can use this to maximize the potential score and place the rolls of the dice into the correct slots.

Example

Consider these dice rolls:

2-3-2-6-1 1-3-1-1-1 4-3-4-6-4 2-3-5-5-6 6-3-5-3-5 1-4-4-6-1

There are a number of ways that these rolls can be used to fill the slots on the upper half of the score sheet. Here are two of them:



Roll					Score		Roll					Score
1	3	1	1	1	4	ones	1	4	4	6	1	2
2	3	2	6	1	4	twos	2	3	2	6	1	4
6	3	5	3	5	6	threes	1	3	1	1	1	3
4	3	4	6	4	12	fours	2	3	5	5	6	0
2	3	5	5	6	10	fives	6	3	5	3	5	10
1	4	4	6	1	6	sixes	4	3	4	6	4	6
Sub Total					42		Sub Total					25
35 point bonus for >= 63?					0		35 point bonus for >= 63?					0
Total					42		Total					25

Obviously there are other potential orderings and totals, and that at least based on the above, a score of 42 is better than a score of 25. Remember that if the sub-total is at least 63, then there is a 35 point bonus, although this does not occur in this example.

Write a program to accept one or more sets of six rolls of five dice, and to display the maximum number of points that can be scored given the above rules and the given set of dice rolls. All that is necessary is to display this maximum score; it is not necessary to explain the placing of the rolls into the slots.

Input

There will be multiple sets of input. Each set contains six lines of data in the format described above: five single-digit numbers with dashes between each. Thus each line contains exactly 9 characters followed by the end of the line. For each set of six lines, display the maximum possible score. There will be a single line of all zeros (`0-0-0-0-0`) after the last set of rolls.

Output

For each input data set display the input set case number $(1, 2, \ldots)$ and the maximum possible score using the input rolls. Use the format shown in the sample output.

Sample Input

```
2-3-2-6-1
1-3-1-1-1
4-3-4-6-4
2-3-5-5-6
6-3-5-3-5
1-4-4-6-1
2-2-3-2-2
4-6-6-6-5
5-4-5-5-5
3-3-1-1-3
1-2-3-2-3
4-3-3-3-3
0-0-0-0-0
```

Sample Output

```
Case 1: 42
Case 2: 60
```