L – The Game

Time limit: 1 s *Memory limit: 256 MiB*

Vladimir is the loneliest child in the neighbourhood. No other kid likes to play with him. His parents decided to cheer him up so they bought him a card game called *The Game*. This card game is for up to 5 players, but it can also be played in the *solo* (i.e. single-player) mode.

The package contains 98 *regular* playing cards that are labeled by integers $2, 3, \ldots, 99$. In addition to these, there are 4 special *direction* cards. Two of them are labeled with the number 1 (followed by an up arrow) and the other two are labeled with 100 (followed by a down arrow).

In the initial phase of the game, the pile of regular cards is shuffled and placed face down on the table – this will be the *draw* pile. The four direction cards are placed in a column; the two cards labeled 1 have to be at the top. There should also be enough space on the right-hand side of each direction card – this is where regular cards will be laid during the play. The card labeled 1 initiates an *ascending row*, while a card labeled 100 initiates a *descending row*. In the solo mode, the player draws the top 8 cards from the draw pile, one by one, and puts them in his hand.

After the initial phase the game starts. On each turn the player has to play two cards from his hand according to the following rules:

- A card may be placed at the end of an ascending row if it is larger than the last (i.e. right-most) card in the row.
- A card may be placed at the end of a descending row if it is smaller than the last card in the row.
- A card with a smaller label may be placed at the end of an ascending row, or a card with a larger label may be place at the end of a descending row, if the absolute difference between its value and the value of the last card in the row is exactly 10. This move is called the *backwards trick*. Note that because of this extra rule, the values of the cards in an *ascending row* are not necessarily ascending (and similarly, the values of the cards in a *descending row* are not necessarily descending).

After playing two cards from the hand, the player should draw two new cards from the draw pile, one by one. This concludes his turn. If the draw pile is empty, he continues playing in the same way without drawing new cards. The game ends when the player has no cards left in his hand (in that case the player *beats the game*) or if he cannot play any of the remaining cards in his hand (in that case the player has *lost the game*).

Example: Suppose that the player's initial hand (i.e. the first 8 cards which he has drawn) is:

```
69, 17, 59, 32, 31, 77, 87, 89
```

He may decide to play the card 89 (putting it in the first descending row) and the card 17 (putting it in the second ascending row). The state of all four rows after the move is:

```
1 ->
1 -> 17
100 <- 89
100 <-
```

Then he has to pick up two more cards from the draw pile – suppose these two cards are 84 and 3 – and his hand becomes:

```
69, 59, 32, 31, 77, 87, 84, 3
```

In the second turn he might want to play the card 3 (in the first ascending row) and card 87 (in the first descending row, after card 89). The state of all four rows after the move:

```
1 -> 3
1 -> 17
100 <- 89, 87
100 <-
```

Vladimir played the game for a few times and he could not always beat the game. Since he hates losing the game, you should write a computer program that will inspect the draw pile and predict the outcome of the game. This will help Vladimir to decide whether he wants to play it or not.

You should also know that Vladimir is a very logical and predictable person. He plays according to the following rules.

- When he draws a card, he places it in his hand on the far-right side.
- He will always play a card from his hand according to his list of priorities:
    1. If one or more cards allow him to do the backward trick, he will use the leftmost such card. If that card can be used for the backward trick in different rows, he will use the top-most amongst these rows.
    2. Otherwise, he plays a card in the regular way. He will select the card to play, and the row in which to put it, in such a way as to minimize the absolute value of the difference between the value of the card that is being played and the last card in the row. If several cards attain the minimum, he will use the left-most amongst these cards. Finally, if there are several choices of where to play this card, he will choose the top-most row.

Your program should find the final state of the game.

Input data

The first (and only) line of the input contains 98 space-separated integers, i.e. some permutation of the set $\{2, 3, \ldots, 99\}$ that represents the initial draw pile. The cards are listed in order from top to bottom of the draw pile.

Output data

The output contains six lines. The first four lines describe the four rows of cards on the table. The fifth line lists the cards that remained in the player's hand (if any) while the last line lists the cards that remained in the draw pile (if any). Print an empty line in case of an empty list. Cards in the four rows and in the hand should be ordered from left to right, while the cards in the last line, which represents the remainder of the draw pile, should be ordered from top to bottom as in the input data. See also the sample outputs.

Examples

Input

```
96 69 40 94 35 7 53 88 10 89 47 37 16 61 24 46 90 6 33 25 63 73 26 81 2 45
    77 75 48 57 66 34 59 92 44 11 31 18 9 52 91 50 8 98 5 64 86 62 83 4 19
    3 27 97 28 36 23 76 58 30 38 12 39 78 41 56 80 67 70 99 13 42 17 49 84
      22 32 29 54 71 51 74 79 95 72 15 87 21 65 68 60 85 55 43 93 20 82 14
```

Output

```
1 7 10 16 6 9 11 18 31 62 64 83 86 91 92 97 98 99
1 2 5 8 19 23 27 28 30 36 38 39 41 56 58 67 70 76 78 80 84 74 79 95
100 96 94 89 88 69 61 53 47 46 40 37 35 33 26 25 24 34 44 42 22 32 29 17
                                                              13 12 4 3
100 90 81 77 75 73 66 63 59 57 52 50 48 45 21 15
49 54 71 51 72 87 65 68
60 85 55 43 93 20 82 14
```

Input

```
87 31 58 56 82 93 9 68 65 41 26 64 3 11 5 84 24 46 16 30 14 85 52 12 91 75
   96 17 47 37 76 69 78 49 25 28 48 81 95 63 34 43 27 74 80 62 53 83 40 71
     72 35 23 21 51 66 55 61 67 32 38 29 60 39 4 18 20 77 7 94 59 42 79 10
      92 97 57 2 86 33 89 90 88 19 22 99 45 44 73 70 50 6 15 98 54 13 36 8
```

Output

```
1 9 11 16 24 14 17 26 28 30 31 34 62 74 78 80 81 71 72 83 95 96 97 99
1 3 5 12 25 27 29 38 39 42 59 60 66 67 57 77 79 86 89 90 92 94 98 88
100 93 87 82 68 65 64 58 56 46 41 37 47 43 53 51 61 55 45 44 33 22 20 19
                                                          15 13 10 8 6
100 91 85 84 76 75 69 63 52 49 48 40 35 32 23 21 18 7 4 2
73 70 50 54 36
```

Note. The inputs and outputs contain lines that do not fit on an A4 page. If a line starts with one or more spaces, it is in fact continuation of the previous line. The actual data (see the judge system) does not contain these line-breaks.