Overview: Given a deck of cards, find a series of moves such that you end with the best hand.

Description: Since recovering from his last battle, Hiiro is once again competing in the back-alley robot competitions. In addition to defeating other robots in the pen, each competitor also receives a special monthly challenge that must be completed to ensure advancement to the next round of robot competitions. In May's challenge, Hiro is given a deck of *n* cards -- each with a unique integer value between 1 to *n* -- from which he can draw cards, and he needs to output the most optimal final hand when our deck runs out.

Each person starts with 0 cards in their chosen hand and can add cards in this matter:

1) Draw 3 cards from the deck, but of the 3, choose 2 to discard.
2) Draw 2 cards from the deck, but of the 2, choose 1 to discard.

Similarly, evaluating the superiority of one possible move over another is defined by:

1) Compare the largest numbers of each hand. The hand with the larger largest number is better.
2) If the two largest numbers are the same, compare the second largest numbers of each hand. The hand with the larger second largest number is better.
3) If those two are the same, compare the third largest numbers of each hand and so on.

Can you use your skills to help Hiro and his robot make it past this month's challenge?

Filename: deck.{java, cpp, c, cc, py}

Input: The input consists of *n*+1 lines.
The first line contains *n*, the number of cards.
The next *n* lines each contain a single number per line indicating the order of the deck. A number on an earlier line is closer to the top of the deck.

Output: In decreasing order, output the best possible final hand with a number per line.

Assumptions: $2 \le n \le 1000$

Sample Input #1:
```
4
1
2
3
4
```

Sample Output #1:
```
4
2
```

Sample Input #2:
```
5
5
1
2
3
4
```

Sample Output #2:

```
5
4
```