Boring Card Game

`cards.c`, `cards.C`, `cards.java`

The Neal Stephenson's novel Cryptonomicon includes a cryptographic algorithm based on a deck of playing cards. It is emphasized that a proper shuffling is crucial for the cipher security. In this problem, we want to demonstrate the importance of randomness in cryptography by describing a card game that does not employ shuffling and is therefore foreseeable.

The game is a modification of poker, where not only all the cards are visible to everyone, but the players have no influence on the course of the game at all. Pretty boring, isn't it?

A *game session* is composed of (possibly) many games and is played by N players. For simplicity, we will assume that the players are sitting in a row and are numbered $1 \dots N$ from left to right. The deck contains exactly $5 \times N$ cards numbered $1, 2, \dots, 5N$.

At the beginning of each game, cards are dealt in three dealing rounds. First, two cards are dealt to each player in the left-to-right order. That is, player 1 gets the two top-most cards, then player 2 gets the next 2 cards and so on until the last player N gets his/her cards. In the second round, this step is repeated and every player gets another two cards in the same manner. Finally, each player gets one more card.

The player who receives all the five cards with the smallest numbers $(1, 2, 3, 4, 5$, not necessarily in this order) is the winner of the whole game session.

If nobody wins, the cards are collected and a new game is started. The cards are collected from the players from right to left and the cards of one player are always collected one-by-one in the reverse order then they were dealt. Each card is placed on top of the deck, another card onto it, and so on. That is, the top of the deck will contain cards of player number 1 and the six top-most cards will be the cards at positions 1, 2, $2N+1$, $2N+2$, $4N+1$, 3 in the original deck.

For example, in the game of two players the initial deck contains ten cards: A, B, C, D, E, F, G, H, I, J. In the first dealing round, player 1 gets the cards A and B, player 2 gets C and D. Then E and F is dealt to player 1, G and H to player 2, then I to player 1, and finally J to player 2. When collecting, the cards of the player 2 go first in the order J, H, G, D, C. Then we continue with player 1's cards I, F, E, B, and A. Since the cards are put onto the deck bottom-top, the final order of the cards after one game is A, B, E, F, I, C, D, G, H, and J.

Write a program that will determine the outcome of a game session so that you can spoil the game to its players.

Input Specification

The input contains several game sessions. Each session is described by two lines. The first line contains the number N, $1 \leq N \leq 1000$. The second line contains the card numbers $1 \dots 5N$ in the order from the top of the deck to the bottom. Every two consecutive numbers on this line are separated by a single space. Each number will occur exactly once on that line.

The last decription is followed by a line containing one zero.

Output Specification

For each game session, output exactly one line. If no player ever wins, print "`Neverending game.`", otherwise output the sentence "`Player` P `wins game number` G`.`", where P is the player number and G is the number of the first game won (the first game is numbered 1). Please note that the result may exceed 2^{32} but it will always be less than 2^{63}.

Sample Input

```
2
2 3 9 7 4 8 5 1 10 6
2
2 6 9 7 4 8 5 1 10 3
5
16 12 18 11 20 15 19 24 8 6 25 1 7 22 14 2 3 10 13 17 4 5 21 9 23
0
```

Output for Sample Input

```
Player 1 wins game number 3.
Neverending game.
Player 2 wins game number 153.
```