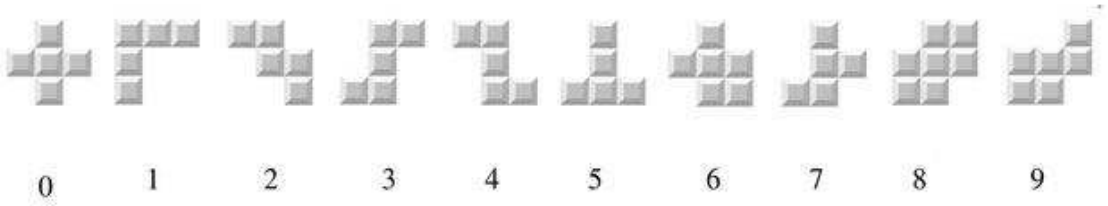
# Problem I
*Tetris*

Input File: I.in
Output File: standard output
Time Limit: *2.5* seconds (C/C++)
Memory Limit: *512* megabytes

Sonya is looking through her old toys. Among cubes and dolls she found the old video game Tetris she loved playing with. This Tetris is quite unusual, it has pieces of **10** different shapes and a grid of width **3** and height **10**. You can imagine the game as an infinite stream of incoming pieces in which a sequence **a** is repeated continuously.

The girl decides to play Tetris. Before a piece falls down, Sonya can rotate it by any angle divisible by **90** degrees, but she can't flip it over. If all the cells in a row are covered, the row disappears, and all the rows on top of it fall down, emptying the row above them. The game is over when there is no more room for the next piece on the grid.

Sonya is smarter now, and she wants to maximize the number of incoming pieces before the game is over. The game might also never end. In this case you should tell Sonya not to start playing.




**Input**
The first line contains the integer **n** ($1 \le n \le 50$) - the length of the sequence **a**
The second line contains **n** integers $a_i$ ($0 \le a_i \le 9$) - the elements of the sequence **a**

**Output**
Print one number - the maximum number of pieces that will fall down before game termination or -1, if it is possible to play and get an infinite number of pieces falling down.

| Sample input | Sample output |
|---|---|
| 1<br>0 | 4 |
| 2<br>3 4 | 12 |
| 3<br>5 1 1 | -1 |