

Peculiar Permutivores

On the gentle plains of Bytelandia, the mysterious ruffalo wander in herds solving combinatorics problems and searching for their next meals. Finding suitable food is becoming more and more difficult, as over the years the ruffalo have developed a curious diet consisting solely of permutations. To make matters worse, roughly half of all existing ruffalo lack the enzyme that would allow their stomachs to decompose permutations into cycles, and the other half are allergic to explicitly written fixed points.

The two great ruffalo leaders, Wildthings and Nowyouseemee, have joined forces to build a device that can automatically transform permutations into a notation that everyone can digest and enjoy (but not necessarily in that order). The bright young programmer Zodiac was assigned the task of building the device, but in a tragic freak accident he swallowed a fixed point and died shortly thereafter. Zodiac's assistants found a single sheet of paper lying next to his body on the laboratory floor with just eight symbols written on it: "+-<>[],". Out of respect for Zodiac's dying wish, the Ruffalo High Council passed a decree that the device must be built entirely in brainf**k, no matter what the cost. Nowyouseemee has personally offered an unprecedented reward of 54 Gifts of Heaven to anyone who can successfully complete the task. While nobody has ever seen a Gift of Heaven, it is said that they come in magical bright green boxes that become a little faded after a day, and a little more faded after a week.

Note: You can use any programming language you want, as long as it is brainf**k. If you would like to submit in other languages, please see the [tutorial](#) version.

Input

The first line contains an integer T ($1 \leq T \leq 5000$). Then follow $2T$ lines, representing T test cases. The first line of each test case contains an integer N ($1 \leq N \leq 9$), and the second line contains a permutation of $[1..N]$ as a space-separated list of N integers. Every line ends with a single newline character (ASCII 10), including the last line.

Output

T lines containing the disjoint cycle decomposition of the corresponding permutation. For the identity permutation, print "e" without quotes. If there is more than one correct answer, print any of them.

Example

Input:

```
5
3
1 2 3
3
2 1 3
4
2 1 4 3
9
3 8 9 4 1 7 6 2 5
```

9
3 8 9 4 1 7 6 2 5

Output:

e
(1 2)
(1 2)(3 4)
(1 3 9 5)(2 8)(6 7)
(2 8)(9 5 1 3)(7 6)

Additional Info

There are two randomly generated data sets, one with $T=5000$ and the other with $T=500$. The average value of N in each data set is approximately 6.36.

The custom judge reports your source code length in parentheses after the judge result (when available at the time of judge termination). Only valid BF commands are counted. Also, if you don't get AC, you will be told on which data set you first failed (0-indexed). Thanks to [\(Tjandra Satria Gunawan\)](#)(曾毅昆) and [Bin Jin](#), who wrote judges using the same or similar ideas. The master judge C++ source code is available [here](#), and assumes that the test case judge reports source length as score.

For assessing the correctness of program output, the custom judge works just the same as the standard "Ignores extra whitespaces" judge, except that it allows any valid cycle decomposition. In case you don't understand how the standard judge works, this means that e.g. "(1 2)" and "(1 2) (3 4)" would be judged as wrong for the second and third example cases respectively, but printing ten spaces instead of the single space in "(1 2)" is perfectly fine.

My solution at the time of publication has 334 bytes (golfed) and runs in 0.47s with 1.6M memory footprint.

Update: Reduced source to [292](#) as of 2014-04-30.