

Permutation Cycle Decomposition

Note: This is the companion problem to the language-restricted [Peculiar Permutivores](#). The constraints here are higher and the cluster is faster to allow better speed comparison, but otherwise it is the same problem.

Given some permutations, please print the unique cycle decomposition (up to ordering and rotations), excluding fixed points, and using the symbol "e" to denote the empty product.

Input

The first line contains an integer T ($1 \leq T \leq 50000$). Then follow $2T$ lines, representing T test cases. The first line of each test case contains an integer N ($1 \leq N \leq 50$), and the second line contains a permutation of $[1..N]$ as a space-separated list of N integers.

Output

T lines containing the disjoint cycle decomposition of the corresponding permutation. Any correct answer is acceptable.

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 $T=50000$ and the other with $T=5000$. The average value of N in each data set is approximately 26.5.

Constraints are set to allow BF to pass without allowing easy 0.00s in C. My BF solution at the time of publication has 476 bytes and runs in 24.50s with 1.9M memory footprint. My C solution runs in 0.02s.

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.