B. Make It Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes

There is a matrix A of size $n \times n$ where $A_{i,j} = j$ for all $1 \le i, j \le n$.

In one operation, you can select a row and reverse any subarray* in it.

Find a sequence of at most $2n$ operations such that every column will contain a permutation† of length n.

It can be proven that the construction is always possible. If there are multiple solutions, output any of them.

*An array a is a subarray of an array b if a can be obtained from b by deleting zero or more elements from the beginning and zero or more elements from the end.

†A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array), and $[1,3,4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains one integer n ($3 \le n \le 5000$) — denoting the number of rows and columns in the matrix.

It is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, on the first line, print an integer k ($0 \le k \le 2n$), the number of operations you wish to perform. On the next lines, you should print the operations.

To print an operation, use the format "$i\ l\ r$" ($1 \le l \le r \le n$ and $1 \le i \le n$) which reverses the subarray $A_{i,l}, A_{i,l+1}, \ldots, A_{i,r}$.

Example

input

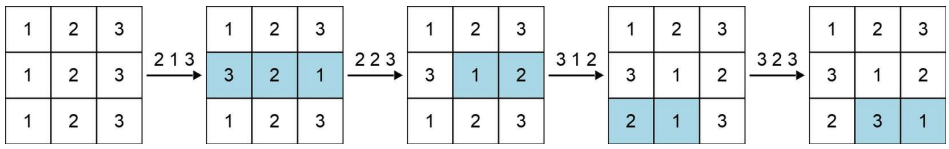
```
2
3
4
```

output

```
4
2 1 3
2 2 3
3 1 2
3 2 3
5
2 1 4
3 1 3
3 2 4
4 3 4
4 1 2
```

Note

In the first test case, the following operations are a valid solution:



- Tutorial (en)