In his August 1986 column for *Isaac Asimov's Science Fiction Magazine* (p.100), Martin Gardner presented an interesting problem:

> *Now for a curious little combinatorial puzzle involving the twelve numbers on the face of a clock. Can you rearrange the numbers (keeping them in a circle) so no triplet of adjacent numbers has a sum higher than 21? This is the smallest value that the highest sum of a triplet can have.*
>
> *I know of no procedure for finding such a permutation, but there must be a way to write a computer program that will print all such permutations in a reasonable time.*
>
> Martin Gardner's source: Dean S. Clarck, "A combinatorial Theorem on Circulant Matrices". **Amer. Math. Monthly**, December 1985, pp. 725 ff.

For our purposes, we will not print all such permutations, but simply count them. Also, to allow for a variety of cases, we will accept as input (1) how many numbers (N) there are on our "clock face" (meaning that we will have numbers in the range from 1 to N), and (2) the largest allowed triplet sum. Hence, for the problem as posed by Martin Gardner, the input would be

```
12 21
```

A number of clock faces will be equivalent based on rotations of a circular permutation — for instance, "123", "231", and "312" are all equivalent, differing only in the starting point. Consequently you are to eliminate this duplication. There are also permutations whose equivalence is based on mirror images (such as "123" and "321"). Eliminate that duplication as well.

Write a program that will accept a number of problems from input file.

Input

The first file entry will specify the number of problems posed in the file. Subsequent entries will be pairs of numbers: (1) how many entries there are on a "clock face" (up to 13), and (2) the largest allowed triplet sum.

Output

The program will report the number of *unique* circular permutations (eliminating permutations equivalent on rotation and/or reflection) meeting the constraint that no triplet found in the circular permutation sums to more than the specified maximum triplet sum.

Note: judging will be based on number pairs that either have no permutations meeting the criterion or have the minimum non-zero number of permutations meeting it.

Specifically, the program will read one integer, followed by pairs of integers from the file, as described above. It will print to the standard output its results (echoing back the input data and giving the number of unique permutations meeting the constraint) as three lines of text.

Sample Input

```
2
6 11
12 21
```

Sample Output

```
Permutation size:    6
Maximum triplet sum: 11
Valid permutations:  1

Permutation size:    12
Maximum triplet sum: 21
Valid permutations:  261
```