CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS

Košice, Slovak Republic
30 June – 6 July 2002

A decorative fence

Input file:	`fence.in`	**100 points**
Output file:	`fence.out`	**Time limit:** 1 s
Source Code:	`fence.pas/.c/.cpp`	**Memory limit:** 1 MB

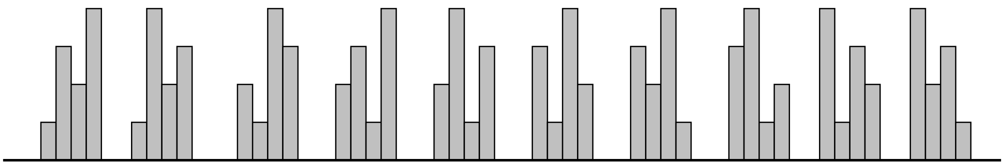
Richard just finished building his new house. Now the only thing the house misses is a cute little wooden fence. He had no idea how to make a wooden fence, so he decided to order one. Somehow he got his hands on the ACME[1] Fence Catalogue 2002, the ultimate resource on cute little wooden fences. After reading its preface he already knew, what makes a little wooden fence cute.

A wooden fence consists of N wooden planks, placed vertically in a row next to each other. A fence looks cute if and only if the following conditions are met:

- The planks have different lengths, namely 1, 2, …, N plank length units.
- Each plank with two neighbors is either larger than each of its neighbors or smaller than each of them. (Note that this makes the top of the fence alternately rise and fall.)

It follows, that we may uniquely describe each cute fence with N planks as a permutation $a_1, \ldots, a_N$ of the numbers $1, \ldots, N$ such that $(\forall i; 1 < i < N)\,(a_i - a_{i-1}) * (a_i - a_{i+1}) > 0$ and vice versa, each such permutation describes a cute fence.

It is obvious, that there are many different cute wooden fences made of N planks. To bring some order into their catalogue, the sales manager of ACME decided to order them in the following way: Fence A (represented by the permutation $a_1, \ldots, a_N$) is in the catalogue before fence B (represented by $b_1, \ldots, b_N$) if and only if there exists such i, that $(\forall j < i)\, a_j = b_j$ and $(a_i < b_i)$. (Also to decide, which of the two fences is earlier in the catalogue, take their corresponding permutations, find the first place on which they differ and compare the values on this place.) All the cute fences with N planks are numbered (starting from 1) in the order they appear in the catalogue. This number is called their *catalogue number*.



All cute fences made of $N = 4$ planks, ordered by their catalogue numbers.

After carefully examining all the cute little wooden fences, Richard decided to order some of them. For each of them he noted the number of its planks and its catalogue number. Later, as he met his friends, he wanted to show them the fences he ordered, but he lost the catalogue somewhere. The only thing he has got are his notes. Please help him find out, how will his fences look like.

[1] A Company Making Everything

Input

The first line of the input file contains the number K ($1 \leq K \leq 100$) of input data sets. K lines follow, each of them describes one input data set.

Each of the following K lines contains two integers N and C ($1 \leq N \leq 20$), separated by a space. N is the number of planks in the fence, C is the catalogue number of the fence.

You may assume, that the total number of cute little wooden fences with 20 planks fits into a 64-bit signed integer variable (`long long` in C/C++, `int64` in FreePascal). You may also assume that the input is correct, in particular that C is at least 1 and it doesn't exceed the number of cute fences with N planks.

Output

For each input data set output one line, describing the C-th fence with N planks in the catalogue. More precisely, if the fence is described by the permutation $a_1, \ldots, a_N$, then the corresponding line of the output file should contain the numbers a_i (in the correct order), separated by single spaces.

Example

Input	Output
2	1 2
2 1	2 3 1
3 3	