Increasing Subsequences

A sequence **p(1)**, **p(2)**, ..., **p(N)** consisting of numbers 1, 2, ..., **N** is called a permutation if all elements in the sequence are different.

It is said that a permutation **p** contains increasing subsequence of **k** elements when there are numbers $1 \le \mathbf{i_1} < \mathbf{i_2} < ... < \mathbf{i_k} \le N$ such that $\mathbf{p(i_1)} < \mathbf{p(i_2)} < ... < \mathbf{p(i_k)}$.

When a permutation **p** contains an increasing subsequence consisting of **B** elements and does not contain an increasing subsequence consisting of **B+1** elements then the number **B** is called the degree of increase of this permutation.

You need to write a program which being given a number **N** calculates the number of permutations whose degree of increase is **B**. Since the number of such permutations might be quite big, it is necessary to calculate its remainder of integer division by 1 000 000 000.

Input

First line of input contains integer **T** ($1 \le \mathbf{T} \le 60$) - the number of testcases. Then descriptions of **T** testcases follow.

The description of the testcase consists of one line. The line contains two integer numbers **N** and **B** ($1 \le \mathbf{N} \le 40$, $1 \le \mathbf{B} \le 5$) separated by one or more spaces.

Output

For each testcase in the input your program should output one line. This line should contain one integer number which is the remainder of integer division by 1 000 000 000 of the number of permutations whose degree of increase is **B**.

Example

Input:
1
3 2

Output:
4