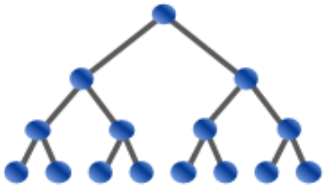

USACO 2020 US Open Contest, Platinum
Problem 2. Exercise

Farmer John has come up with a new morning exercise routine for the cows (again)!

As before, Farmer John's N cows ($1 \le N \le 7500$) are standing in a line. The i-th cow from the left has label i for each $1 \le i \le N$. He tells them to repeat the following step until the cows are in the same order as when they started.

- Given a permutation A of length N, the cows change their order such that the i-th cow from the left before the change is A_i-th from the left after the change.

For example, if $A = (1, 2, 3, 4, 5)$ then the cows perform one step and immediately return to the same order. If $A = (2, 3, 1, 5, 4)$, then the cows perform six steps before returning to the original order. The order of the cows from left to right after each step is as follows:

- 0 steps: $(1, 2, 3, 4, 5)$
- 1 step: $(3, 1, 2, 5, 4)$
- 2 steps: $(2, 3, 1, 4, 5)$
- 3 steps: $(1, 2, 3, 5, 4)$
- 4 steps: $(3, 1, 2, 4, 5)$
- 5 steps: $(2, 3, 1, 5, 4)$
- 6 steps: $(1, 2, 3, 4, 5)$

Compute the product of the numbers of steps needed over all $N!$ possible permutations A of length N.

As this number may be very large, output the answer modulo M ($10^8 \le M \le 10^9 + 7$, M is prime).

Contestants using C++ may find the following code from KACTL helpful. Known as the Barrett reduction, it allows you to compute $a\%b$ several times faster than usual, where $b > 1$ is constant but not known at compile time. (we are not aware of such an optimization for Java, unfortunately).

```
#include <bits/stdc++.h>
using namespace std;

typedef unsigned long long ull;
typedef __uint128_t L;
struct FastMod {
	ull b, m;
	FastMod(ull b) : b(b), m(ull((L(1) << 64) / b)) {}
	ull reduce(ull a) {
		ull q = (ull)((L(m) * a) >> 64);
		ull r = a - q * b; // can be proven that 0 <= r < 2*b
		return r >= b ? r - b : r;
	}
};
FastMod F(2);

int main() {
	int M = 1000000007; F = FastMod(M);
	ull x = 10ULL*M+3;
	cout << x << " " << F.reduce(x) << "\n"; // 10000000073 3
}
```

INPUT FORMAT (file exercise.in):

The first line contains N and M.

OUTPUT FORMAT (file exercise.out):

A single integer.

SAMPLE INPUT:

```
5 1000000007
```

SAMPLE OUTPUT:

```
369329541
```

For each $1 \le i \le N$, the i-th element of the following array is the number of permutations that cause the cows to take i steps: $[1, 25, 20, 30, 24, 20]$. The answer is $1^1 \cdot 2^{25} \cdot 3^{20} \cdot 4^{30} \cdot 5^{24} \cdot 6^{20} \equiv 369329541 \pmod{10^9 + 7}$.

Note: This problem has an expanded memory limit of 512 MB.

SCORING:

- Test case 2 satisfies $N = 8$.
- Test cases 3-5 satisfy $N \le 50$.
- Test cases 6-8 satisfy $N \le 500$.
- Test cases 9-12 satisfy $N \le 3000$.
- Test cases 13-16 satisfy no additional constraints.

Problem credits: Benjamin Qi

Contest has ended. No further submissions allowed.