Splits

For a permutation $p = p[0]\ p[1]\ p[2]\ \ldots\ p[n-1]$ of the numbers $1, 2, 3, \ldots, n$ we define a *split* as a permutation q which can be obtained by the following process:

1. Select two sets of numbers $A = \{ i_1, i_2, ..., i_k \}$ and $B = \{ j_1, j_2, ..., j_l \}$ such that $A \cap B = \emptyset$, $A \cup B = \{ 0, 1, 2, ..., n-1 \}$, $i_1 < i_2 < ... < i_k$ and $j_1 < j_2 < ... < j_l$
2. The permutation q will be $q = p[i_1]p[i_2]\ldots p[i_k]p[j_1]p[j_2]\ldots p[j_l]$

Moreover, we define $S(p)$ to be the set of all *splits* of a permutation p.

You are given a number n and a set T of m permutations of length n. Count how many permutations p of length n exist such that $T \subseteq S(p)$. Since this number can be large, find it modulo $998\,244\,353$.

Implementation Details

You should implement the following procedure:

```
int solve(int n, int m, std::vector<std::vector<int>>& splits);
```

- n: the size of the permutation
- m: the number of splits
- $splits$: array containing m **pairwise distinct** permutations, the elements of the set T, which is a subset of $S(p)$
- This procedure should return the number of possible permutations modulo $998\,244\,353$.
- This procedure is called exactly once for each test case.

Constraints

- $1 \leq n \leq 300$
- $1 \leq m \leq 300$

Subtasks

1. (6 points) $m = 1$
2. (7 points) $1 \leq n, m \leq 10$
3. (17 points) $1 \leq n, m \leq 18$
4. (17 points) $1 \leq n \leq 30, 1 \leq m \leq 15$

5. (16 points) $1 \leq n, m \leq 90$
6. (16 points) $1 \leq n \leq 300$, $1 \leq m \leq 15$
7. (21 points) No additional constraints.

Examples

Example 1

Consider the following call:

```
solve(3, 2, {{1, 2, 3}, {2, 1, 3}})
```

In this sample, the size of the permutation p is 3 and we are given 2 splits:

- $1\ 2\ 3$
- $2\ 1\ 3$

The function call will return 4 as there are only four permutations p that can generate both of those splits:

- $1\ 2\ 3$
- $1\ 3\ 2$
- $2\ 1\ 3$
- $2\ 3\ 1$

Sample grader

The sample grader reads the input in the following format:

- line 1: $n\ m$
- line $2 + i$: $s[i][0]\ s[i][1]\ \ldots\ s[i][n-1]$ for all $0 \leq i < m$

and outputs the result of the call to `solve` with the corresponding parameters.