Problem 1. Magic Number

(Time Limit: 2 seconds)

Problem Description

Please select continuous digits from the 01234567890 string, which is mapped to the string CELAMGIGOOC, respectively. A magic word will be available by using specific array selection and operation process as follows.

(1). Select continuous digits.

(2). Reverse the selected digits.

(3). Take the absolute value of the difference value from (1) and (2).

(4). Reverse the digits again.

(5). Take the sum value from (3) and (4), then multiply it by 40.

(6). Take the digits from (5) and map it to the string CELAMGIGOOC to get a magic word.

Input Format

The first line contains an integer $t \le 20$ indicating the number of test cases. Each case consists of exactly two lines. The first line is an integer n ($3 \le n \le 7$) indicating the number of integers in the second line. The second line consists of n continuous digits selected from 01234567890.

Output Format

Output the magic word.

Example

Sample Input:	**Sample Output:**
2 3 123 5 56789	MAGIC MAOGICC

Problem 2. Self-iterated Power

(Time Limit: 1 seconds)

Problem Description

Let $F_1(x) = x$, $F_2(x) = x^x$, $F_3(x) = x^{x^x}$ and in general $F_k(x) = x^{F_{k-1}(x)}$ for any integer $k \geq 2$. We call $F_k(x)$ the k-th self-iterated power of x.

For example, $F_1(2) = 2$, $F_2(2) = 4$, $F_3(2) = 16$, $F_4(2) = 65536$, $F_5(2) = 2^{65536}$. When x is 1 or 2, the exact value of $F_k(x)$ % 2^L is easy to obtain, for any positive integer L. How about the case when x is 3?

Technical Specification

- A positive integer k, $1 \leq k \leq 10000$
- A positive integer L, $1 \leq L \leq 30$
- A positive integer n, $1 \leq n \leq 20$

Input Format

The first line of the input contains an integer n indicating the number of test cases. Each test case contains one line with two positive integers k and L, separated by a space.

Output Format

For each test case, output the exact value of $F_k(3)$ % 2^L on one line.

Example

Sample Input:	Sample Output:
4 1 4 2 4 2 5 3 5	3 11 27 27

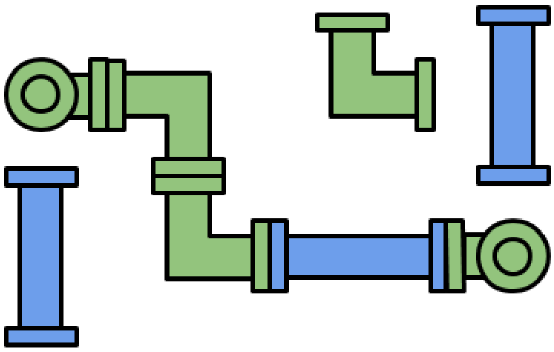
Problem 3. Popeye Loves Pipes

(Time Limit: 5 second)

Problem Description

Popeye has a bunch of pipes. Whenever Olive – Popeye's girlfriend is angry, she broke the pipe. Then Popeye needs to fix the pipes. They live in a small terminal which looks like an $N*M$ grid. In all cells of the grid except (0, 0) and (N-1, M-1), there is a pipe in either L-shaped or I-shaped, connecting exactly two of the four boundaries of the cell. At cells (0, 0) and (N-1, M-1) there are two L-shaped pipe

Olive has broken some pipes. To repair the entire pipe system, Popeye needs to buy some I-shaped pipes and L-shaped pipes. An I-shaped pipe costs ***I*** dollars and an L-shaped pipe costs ***L*** dollars. Moreover, it is free to remove a broken pipe or change the direction of a pipe in a cell. However, it costs ***P*** dollars to remove a healthy pipe.



Please help Popeye to find the minimum cost fixing the pipes so that he can push waters from (0, 0) to (N-1, M-1) without making any floods.

Technical Specification

- The number of test case $T \leq 100$
- $2 \leq N, M \leq 10$
- $0 \leq I, L, P \leq 100$

Input Format

The first line contains an integer *T* indicating the number of test cases. For each test case, first line contains five integers *N*, *M*, *I*, *L*, *P* separated by whitespaces. Then *N* lines follow. Each line contains a string of length *M*. The character I denotes an I-shaped pipe, the character L denotes an L-shaped pipe and the character X denotes a broken pipe. The pipes in (0, 0) and (*N*-1, *M*-1) are always L-shaped pipes.

Output Format

For each test case, output the minimum cost to guarantee a path form (0, 0) to (*N*-1, *M*-1).

Example

Sample Input:	**Sample Output:**
2 5 5 1 1 1 LXXXX LIIIL XXXXX IIIII XXXXL 2 4 1 1 1 LLLI ILIL	1 0

Problem 4. Two Pylons

(Time Limit: 2 seconds)

Problem Description

There are n buildings on a 2D plane. The location of building i is (x_i,y_i). In order to supply electricity to a building, we have to build a pylon. A pylon may supply electricity to arbitrarily many buildings, and we can construct a pylon at any location. However, the distance between the building and the pylon must be no more than some given constant r, otherwise we cannot transfer electricity from the pylon to the building. Due to limited budget, we can only build at most two pylons. Please write a program to determine whether we can transfer electricity to all buildings by only two pylons. In other words, you may choose two locations to build the pylons, and all buildings can reach a pylon within a distance r.

Technical Specification

- $1 \leq T \leq 100$
- $n \leq 20$
- $1 \leq r \leq 300$
- $-100 \leq x_i, y_i \leq 100$ for every $1 \leq i \leq n$
- All numbers in the input are integers. However, you should treat all coordinates and distances as double-precision floating-point numbers.

Input Format

The first line of the input contains an integer T indicating the number of test cases. The first line of each case contains two integers n and r separated by a blank where n is the number of buildings and r is the distance limit. The i-th of the following n lines contains two integers x_i and y_i indicating that the coordinate of the i-th building is (x_i,y_i).

Output Format

For each test case, print "yes" if all buildings can be supplied by two pylons. Otherwise, print "no".

Example

Sample Input:	**Sample Output:**
3 3 1 0 0 3 3 6 6 2 1 0 100 100 0 3 1 0 0 1 1 2 2	no yes yes

Problem 5. Lisa Loves LIS

(Time Limit: 2 seconds)

Problem Description

Lisa is good at combinatorics. She just learnt the concept of longest increasing subsequence (LIS). She loves it so much since LIS is a prefix of her name, LISA!

Computing the length of LIS is too easy for her. She not only knows the $O(N^2)$ dynamic programming algorithm, but also the $O(N\log N)$ approach with binary search.

To challenge herself, she wants to know how many sequences with length n, where each element is an integer between 1 and m, such that the length of LIS of this sequence is k?

For example, if $n = m = k = 5$, the only sequence meet above conditions is [1, 2, 3, 4, 5]. If $n = 3$, $m = k = 2$, there are four sequences meet above conditions, which are [1, 1, 2], [1, 2, 1], [1, 2, 2] and [2, 1, 2].

Technical Specification

- The number of test case $T \leq 1000$
- $1 \leq n, m \leq 12$
- $1 \leq k \leq \min(n, m)$

Input Format

The first line contains an integer T indicating the number of the test cases. For each test case, there are three integers n, m, k separated by a blank in one line.

Output Format

For each test case, output the number of sequences meet above conditions.

Example

Sample Input:	Sample Output:
4 5 5 5	1 4

3 2 2 6 3 3 9 8 6	233 677992