Problem A

AUTOMATIC ANSWER

Last month Alice nonchalantly entered her name in a draw for a Tapmaster 4000. Upon checking her mail today, she found a letter that read:

“Congratulations, Alice! You have won a Tapmaster 4000. To claim your prize, you must answer the following skill testing question.”

Alice’s initial feelings of surprised joy turned quickly to those of dismay. Her lifetime record for skill testing questions is an abysmal 3 right and 42 wrong.

Mad Skills, the leading skill testing question development company, was hired to provide skill testing questions for this particular Tapmaster 4000 draw. They decided to create a different skill testing question to each winner so that the winners could not collaborate to answer the question.

Can you help Alice win the Tapmaster 4000 by solving the skill testing question?

Program Input

The input begins with t ($1 \le t \le 100$), the number of test cases. Each test case contains an integer n ($-1000 \le n \le 1000$) on a line by itself. This n should be substituted into the skill testing question below.

Program Output

For each test case, output the answer to the following skill testing question on a line by itself: “Multiply n by 567, then divide the result by 9, then add 7492, then multiply by 235, then divide by 47, then subtract 498. What is the digit in the tens column?”

Sample Input & Output

INPUT

```
2
637
```

```
-120
```

OUTPUT

```
1
3
```

Problem B

BLACKBOARD BONANZA

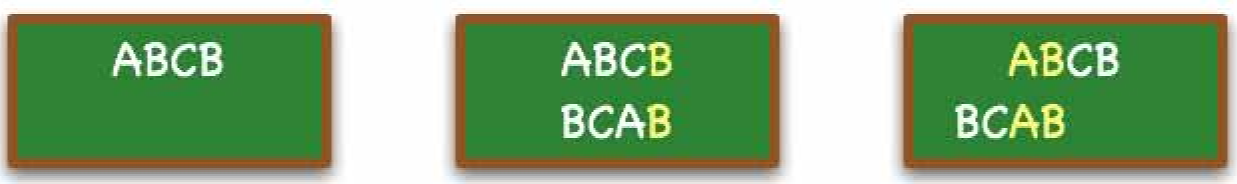
Alice and Bob both have lots of candies but want more. They decide to play the following turn-based game.

First they write some words on a few pieces of paper and put them into a bag so they cannot see the words. Next they decide whose turn is first. The first turn begins with the first player drawing and keeping a piece of paper with the word *A* from the bag and copying *A* onto a blackboard evenly spaced.

Then the second player draws and keeps a piece of paper with the word *B* on it. The current player is to write *B* on the blackboard underneath *A* evenly spaced. The second player receives one candy from the first for each character that matches vertically between *A* and *B*.

Now it is the first player's turn who similarly draws and places word *C* underneath *B* and gains a candy for each of the characters vertically matched between *B* and *C*. The game continues until there are no more words in the bag.

What is the maximum number of candies that one of Alice and Bob will get in a turn?



The game on the second blackboard awards the second player one candy. The game on the third blackboard awards the second player two candies.

Program Input

The first line of the input contains an integer t ($1 \leq t \leq 70$), the number of test cases. Each test case starts with an integer n ($2 \leq n \leq 70$), the number of words in the bag. Then follow n lines containing one word each (in no particular order). Each word will contain between 1 and 70 characters, all of them uppercase letters of English alphabet.

Program Output

For each test case, print a line containing the maximum number of candies either Alice or Bob can get in a single turn.

Sample Input & Output

INPUT

```
2
2
ALICE
BOB
2
ABCB
BCAB
```

OUTPUT

```
0
2
```

Problem C

CALCULATOR CONUNDRUM

Alice got a hold of an old calculator that can display n digits. She was bored enough to come up with the following time waster.

She enters a number k then repeatedly squares it until the result overflows. When the result overflows, only the n most significant digits are displayed on the screen and an error flag appears. Alice can clear the error and continue squaring the displayed number. She got bored by this soon enough, but wondered:

"Given n and k, what is the largest number I can get by wasting time in this manner?"

Program Input

The first line of the input contains an integer t ($1 \leq t \leq 200$), the number of test cases. Each test case contains two integers n ($1 \leq n \leq 9$) and k ($0 \leq k < 10^n$) where n is the number of digits this calculator can display k is the starting number.

Program Output

For each test case, print the maximum number that Alice can get by repeatedly squaring the starting number as described.

Sample Input & Output

INPUT

```
2
1 6
2 99
```

OUTPUT

```
9
99
```

Problem D

DEMANDING DILEMMA

A simple undirected graph is an ordered pair $G = (V, E)$ where V is a non-empty set of vertices, and E is a set of unordered pairs (u, v) where u and v are in V and $u \neq v$. If S is a set, define $|S|$ as the size of S. An incidence matrix M is a $|V|$ x $|E|$ matrix where $M(i, j)$ is 1 if edge j is incident to vertex i (edge j is either (i, u) or (u, i)) and 0 otherwise.

Given an n x m matrix, can it be an incidence matrix of a simple undirected graph $G = (V, E)$ where $|V| = n$ and $|E| = m$?

Program Input

The first line of the input contains an integer t ($1 \leq t \leq 41$), the number of test cases.

Each test case starts with a line with two integers n ($1 \leq n \leq 8$) and m ($0 \leq m \leq n(n-1)/2$). Then n lines containing m integers (0's or 1's) each follow such that the jth number on the ith line is $M(i, j)$.

Program Output

For each test case print "Yes" if the incidence matrix given in the input can be an incidence matrix of some simple undirected graph, otherwise print "No".

Sample Input & Output

INPUT

```
3
3 3
1 1 0
0 1 1
1 0 1
3 1
1
1
0
3 3
1 1 0
1 1 1
```

```
1 0 0
```

OUTPUT

```
Yes
Yes
No
```

Problem E

EXPERIENCED ENDEAVOUR

Alice is given a list of integers by Bob and is asked to generate a new list where each element in the new list is the sum of some other integers in the original list. The task is slightly more involved, as Bob also asks Alice to repeat this several times before giving him the result. Help Alice automate her task.

Program Input

The first line of the input is t ($1 \leq t \leq 10$), the number of cases to follow. Each case is in the following format:

$$
\begin{array}{l}
n\ r \\
a_0\ a_1\ \ldots\ a_{n-1} \\
x_0\ b_{0,0}\ b_{0,1}\ \ldots\ b_{0,x_0-1} \\
x_1\ b_{1,0}\ b_{1,1}\ \ldots\ b_{1,x_1-1} \\
\vdots \\
x_{n-1}\ b_{n-1,0}\ b_{n-1,1}\ \ldots\ b_{n-1,x_{n-1}-1}
\end{array}
$$

Each case begins with the integer n ($1 \leq n \leq 50$), which is the number of elements in the list of integers that Alice is given. The integer r ($1 \leq r \leq 10^9$) is the number of times these operations are to be repeated on a list before returning the result. The values are the nonnegative integers in the original list. Then n lines follow that define how Alice will generate a new list from a previous one. Each of these lines are in the form:

$$x_i\ b_{i,0}\ b_{i,1}\ \ldots\ b_{i,x_i}$$

This line defines the value of the ith element in the new list to be the sum of elements:

$$a_{b_{i,0}}, a_{b_{i,1}}, \ldots, a_{b_{i,x_i-1}}.$$

Program Output

The output consists of t lines, one line for each test case listing the final list of integers modulo 1000 in the form:

$c_0 \;\; c_1 \;.\;.\;.\; c_{n-1}$

Sample Input & Output

INPUT

```
2
2 2
1 2
2 0 1
1 1
2 4
507 692
2 0 1
1 1
```

OUTPUT

```
5 2
275 692
```

Problem F

FEWEST FLOPS

A common way to uniquely encode a string is by replacing its consecutive repeating characters (or "chunks") by the number of times the character occurs followed by the character itself. For example, the string "aabbbaabaaaa" may be encoded as "2a3b2a1b4a". (Note for this problem even a single character "b" is replaced by "1b".)

Suppose we have a string S and a number k such that k divides the length of S. Let $S1$ be the substring of S from 1 to k, $S2$ be the substring of S from $k + 1$ to $2k$, and so on. We wish to rearrange the characters of each block Si independently so that the concatenation of those permutations S' has as few chunks of the same character as possible. Output the fewest number of chunks.

For example, let S be "uuvuwwuv" and k be 4. Then $S1$ is "uuvu" and has three chunks, but may be rearranged to "uuuv" which has two chunks. Similarly, $S2$ may be rearranged to "vuww". Then S', or $S1S2$, is "uuuvvuww" which is 4 chunks, indeed the minimum number of chunks.

Program Input

The input begins with a line containing t ($1 \leq t \leq 100$), the number of test cases. The following t lines contain an integer k and a string S made of no more than 1000 lowercase English alphabet letters. It is guaranteed that k will divide the length of S.

Program Output

For each test case, output a single line containing the minimum number of chunks after we rearrange S as described above.

INPUT

```
2
5 helloworld
7 thefewestflops
```

OUTPUT

```
8
```

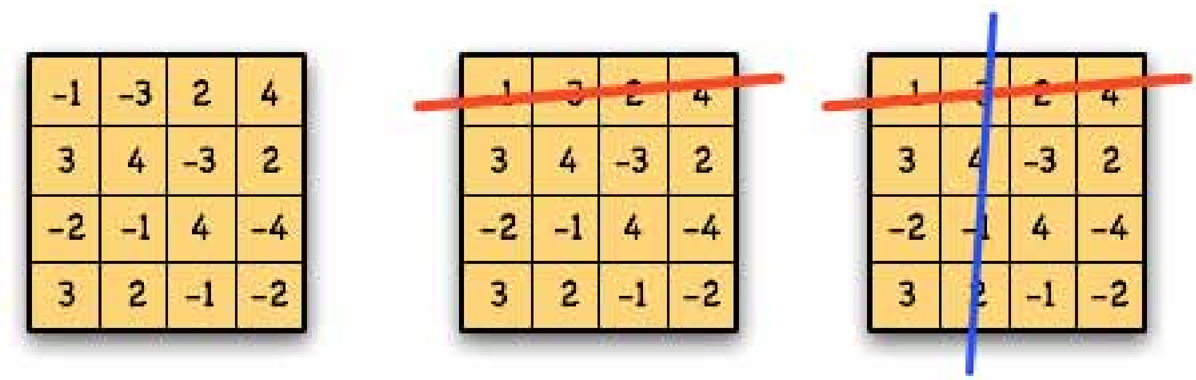
Problem G

GRID GAME

Alice and Bob both have lots of candies but want more. They decide to play the following turn-based game.

They fill an n x n grid M with random integers. Alice begins the game by crossing off an uncrossed row i of the grid. Now it's Bob turn and he crosses off an uncrossed column j of the grid. At the end of Bob's turn, Alice takes the number candies in the ith row and jth column of M, call this value $M(i, j)$, from Bob. (If $M(i, j)$ is negative, then Alice gives $|M(i, j)|$ candies to Bob.) The game continues alternating turns from Alice to Bob until the entire board is crossed off.

What is the largest amount of candies that Alice can win from Bob (or least amount to lose if she cannot win) if both Alice and Bob play optimally?



The beginning of a game between Alice (red) and Bob (blue).

Program Input

The first line of the input contains an integer t ($1 \leq t \leq 20$), the number of test cases. Each test case starts with n ($1 \leq n \leq 8$), the size of the grid. Then follow n lines containing n numbers separated by spaces describing M. We call the jth number on ith line $M(i, j)$ ($-1000 \leq M(i, j) \leq 1000$).

Program Output

For each test case, print the largest amount of candies that Alice can win from Bob. If she cannot win, print the negative number indicating the minimum number of candies she loses.

Sample Input & Output

INPUT

```
3
2
10 10
-5 -5
2
10 -5
10 -5
2
10 -5
-5 10
```

OUTPUT

```
5
5
-10
```

Problem H

HAPLESS HEDONISM

Bob is a world-renowned stick collector. His most prized stick possessions include:

- an Arctic Redwood branch from a hike near Dawson City,
- a Desert Pine stick from a visit to the Grand Canyon, and
- a Chinese Arbour twig from an adventure into Tibet.

Bob collects sticks in a peculiar way. He will only accept a new stick into his collection if its length is exactly length $n + 1$ cm where n is the number of sticks currently in his collection. This implies his collection of n sticks contains exactly one stick of length 1 cm through n cm.

One day Alice visited Bob to inspect his stick collection (upon Bob's insistence of course). Alice wasn't particularly interested in Bob's excessive descriptions and needed a quick conversation changer. Cleverly, she posed the following question to Bob: "If you are allowed to take any 3 sticks from your collection, how many different triangles can you make?"

Can you help Bob answer the question so he can get back to telling Alice about his sticks?

Program Input

The input will begin with t ($1 \leq t \leq 1000$), the number of test cases. Each test case will contain an integer n ($3 \leq n \leq 1000000$), the number of sticks in Bob's collection. (Recall if Bob has n sticks, then he has exactly one stick of each of the lengths from 1 cm through n cm.)

Program Output

For each test case, output on a line the number of different triangles you can make with Bob's sticks. Triangles X and Y are different if there is at least one stick in X that is not in Y. A triangle has area strictly greater than 0.

Sample Input & Output

INPUT

```
3
3
4
10
```

OUTPUT

```
0
1
50
```
