Problem A. Is Prime

Time limit 1000 ms
Mem limit 524288 kB

Again Prime! Boring problem. However, Given an integer `N`, determine if it is a prime a number.

A number is called prime if it is only divisible by itself.

Input

The input will contain one integer `N` (`0 < N < 1000`).

Output

Print `Yes` if the integer is prime, otherwise `No`.

Sample

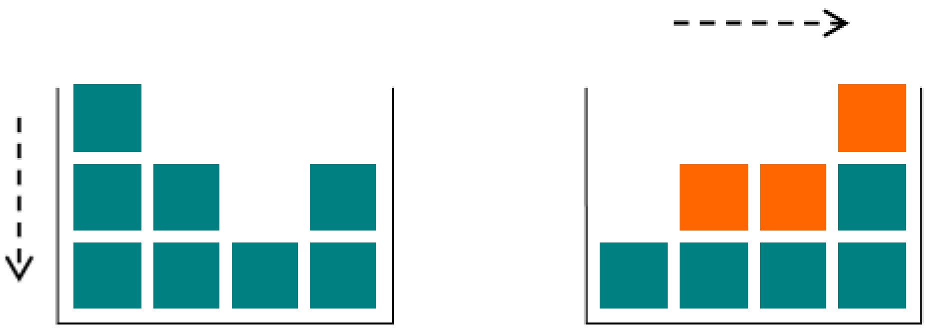
Input	Output
7	Yes

Problem B. Gravity Flip

Time limit 1000 ms
Mem limit 262144 kB

Little Chris is bored during his physics lessons (too easy), so he has built a toy box to keep himself occupied. The box is special, since it has the ability to change gravity.

There are n columns of toy cubes in the box arranged in a line. The i-th column contains a_i cubes. At first, the gravity in the box is pulling the cubes downwards. When Chris switches the gravity, it begins to pull all the cubes to the right side of the box. The figure shows the initial and final configurations of the cubes in the box: the cubes that have changed their position are highlighted with orange.



Given the initial configuration of the toy cubes in the box, find the amounts of cubes in each of the n columns after the gravity switch!

Input

The first line of input contains an integer n $(1 \le n \le 100)$, the number of the columns in the box. The next line contains n space-separated integer numbers. The i-th number a_i $(1 \le a_i \le 100)$ denotes the number of cubes in the i-th column.

Output

Output n integer numbers separated by spaces, where the i-th number is the amount of cubes in the i-th column after the gravity switch.

Sample 1

Input	Output
4 3 2 1 2	1 2 2 3

Sample 2

Input	Output
3 2 3 8	2 3 8

Note

The first example case is shown on the figure. The top cube of the first column falls to the top of the last column; the top cube of the second column falls to the top of the third column; the middle cube of the first column falls to the top of the second column.

In the second example case the gravity switch does not change the heights of the columns.

Problem C. Anton and Letters

Time limit 2000 ms
Mem limit 262144 kB

Recently, Anton has found a set. The set consists of small English letters. Anton carefully wrote out all the letters from the set in one line, separated by a comma. He also added an opening curved bracket at the beginning of the line and a closing curved bracket at the end of the line.

Unfortunately, from time to time Anton would forget writing some letter and write it again. He asks you to count the total number of distinct letters in his set.

Input

The first and the single line contains the set of letters. The length of the line doesn't exceed 1000. It is guaranteed that the line starts from an opening curved bracket and ends with a closing curved bracket. Between them, small English letters are listed, separated by a comma. Each comma is followed by a space.

Output

Print a single number — the number of distinct letters in Anton's set.

Sample 1

Input	Output
`{a, b, c}`	`3`

Sample 2

Input	Output
`{b, a, b, a}`	`2`

Sample 3

Input	Output
`{}`	`0`

Problem D. Jolly Jumpers

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

A sequence of $n > 0$ integers is called a jolly jumper if the absolute values of the difference between successive elements take on all the values 1 through $n - 1$. For instance,

```
1 4 2 3
```

is a jolly jumper, because the absolutes differences are 3, 2, and 1 respectively. The definition implies that any sequence of a single integer is a jolly jumper. You are to write a program to determine whether or not each of a number of sequences is a jolly jumper.

Input

Each line of input contains an integer $n \leq 3000$ followed by n integers representing the sequence. The values in the sequence are at most 300 000 in absolute value. Input contains at most 10 lines.

Output

For each line of input, generate a line of output saying "`Jolly`" or "`Not jolly`".

Sample 1

Input	Output
`4 1 4 2 3` `5 1 4 2 -1 6`	`Jolly` `Not jolly`

Problem F. Harmonic Number

Time limit 2000 ms
Mem limit 65536 kB

In mathematics, the $\mathbf{n^{th}}$ harmonic number is the sum of the reciprocals of the first **n** natural numbers:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n}$$
$$= \sum_{k=1}^{n} \frac{1}{k}$$

In this problem, you are given **n**, you have to find $\mathbf{H_n}$.

Input

Input starts with an integer **T (≤ 10000)**, denoting the number of test cases.

Each case starts with a line containing an integer $\mathbf{n}$ $\mathbf{(1 \le n \le 10^8)}$.

Output

For each case, print the case number and the $\mathbf{n^{th}}$ harmonic number. Errors less than $\mathbf{10^{-8}}$ will be ignored.

Sample

Input	Output
12 1 2 3 4 5 6 7 8 9 90000000 99999999 100000000	Case 1: 1 Case 2: 1.5 Case 3: 1.8333333333 Case 4: 2.0833333333 Case 5: 2.2833333333 Case 6: 2.450 Case 7: 2.5928571429 Case 8: 2.7178571429 Case 9: 2.8289682540 Case 10: 18.8925358988 Case 11: 18.9978964039 Case 12: 18.9978964139

Problem G. Dinner with Emma

Time limit 1000 ms
Mem limit 262144 kB

Jack decides to invite Emma out for a dinner. Jack is a modest student, he doesn't want to go to an expensive restaurant. Emma is a girl with high taste, she prefers elite places.

Munhattan consists of n streets and m avenues. There is exactly one restaurant on the intersection of each street and avenue. The streets are numbered with integers from 1 to n and the avenues are numbered with integers from 1 to m. The cost of dinner in the restaurant at the intersection of the i-th street and the j-th avenue is c_{ij}.

Jack and Emma decide to choose the restaurant in the following way. Firstly Emma chooses the street to dinner and then Jack chooses the avenue. Emma and Jack makes their choice optimally: Emma wants to maximize the cost of the dinner, Jack wants to minimize it. Emma takes into account that Jack wants to minimize the cost of the dinner. Find the cost of the dinner for the couple in love.

Input

The first line contains two integers n, m ($1 \le n, m \le 100$) — the number of streets and avenues in Munhattan.

Each of the next n lines contains m integers c_{ij} ($1 \le c_{ij} \le 10^9$) — the cost of the dinner in the restaurant on the intersection of the i-th street and the j-th avenue.

Output

Print the only integer a — the cost of the dinner for Jack and Emma.

Sample 1

Input	Output
3 4 4 1 3 5 2 2 2 2 5 4 5 1	2

Sample 2

Input	Output

Input	Output
3 3 1 2 3 2 3 1 3 1 2	1

Note

In the first example if Emma chooses the first or the third streets Jack can choose an avenue with the cost of the dinner 1. So she chooses the second street and Jack chooses any avenue. The cost of the dinner is 2.

In the second example regardless of Emma's choice Jack can choose a restaurant with the cost of the dinner 1.

Problem H. Present from Lena

Time limit 2000 ms
Mem limit 262144 kB

Vasya's birthday is approaching and Lena decided to sew a patterned handkerchief to him as a present. Lena chose digits from 0 to n as the pattern. The digits will form a rhombus. The largest digit n should be located in the centre. The digits should decrease as they approach the edges. For example, for $n = 5$ the handkerchief pattern should look like that:

```
          0
        0 1 0
      0 1 2 1 0
    0 1 2 3 2 1 0
  0 1 2 3 4 3 2 1 0
0 1 2 3 4 5 4 3 2 1 0
  0 1 2 3 4 3 2 1 0
    0 1 2 3 2 1 0
      0 1 2 1 0
        0 1 0
          0
```

Your task is to determine the way the handkerchief will look like by the given n.

Input

The first line contains the single integer n ($2 \le n \le 9$).

Output

Print a picture for the given n. You should strictly observe the number of spaces before the first digit on each line. Every two adjacent digits in the same line should be separated by exactly one space. There should be no spaces after the last digit at the end of each line.

Sample 1

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>2</pre></td><td><pre>   0
  0 1 0
0 1 2 1 0
  0 1 0
    0</pre></td></tr>
</table>

Sample 2

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>3</pre></td><td><pre>   0
    0 1 0
  0 1 2 1 0
0 1 2 3 2 1 0
  0 1 2 1 0
    0 1 0
      0</pre></td></tr>
</table>

Problem I. Social Distance

Time limit 2000 ms
Mem limit 262144 kB

Polycarp and his friends want to visit a new restaurant. The restaurant has n tables arranged along a straight line. People are already sitting at some tables. The tables are numbered from 1 to n in the order from left to right. The state of the restaurant is described by a string of length n which contains characters "1" (the table is occupied) and "0" (the table is empty).

Restaurant rules prohibit people to sit at a distance of k or less from each other. That is, if a person sits at the table number i, then all tables with numbers from $i - k$ to $i + k$ (except for the i-th) should be free. In other words, the absolute difference of the numbers of any two occupied tables must be strictly greater than k.

For example, if $n = 8$ and $k = 2$, then:

- strings "10010001", "10000010", "00000000", "00100000" satisfy the rules of the restaurant;
- strings "10100100", "10011001", "11111111" do not satisfy to the rules of the restaurant, since each of them has a pair of "1" with a distance less than or equal to $k = 2$.

In particular, if the state of the restaurant is described by a string without "1" or a string with one "1", then the requirement of the restaurant is satisfied.

You are given a binary string s that describes the current state of the restaurant. It is guaranteed that the rules of the restaurant are satisfied for the string s.

Find the maximum number of free tables that you can occupy so as not to violate the rules of the restaurant. Formally, what is the maximum number of "0" that can be replaced by "1" such that the requirement will still be satisfied?

For example, if $n = 6$, $k = 1$, $s =$ "100010", then the answer to the problem will be 1, since only the table at position 3 can be occupied such that the rules are still satisfied.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases in the test. Then t test cases follow.

Each test case starts with a line containing two integers n and k ($1 \le k \le n \le 2 \cdot 10^5$) — the number of tables in the restaurant and the minimum allowed distance between two people.

The second line of each test case contains a binary string s of length n consisting of "0" and "1" — a description of the free and occupied tables in the restaurant. The given string satisfy to the rules of the restaurant — the difference between indices of any two "1" is more than k.

The sum of n for all test cases in one test does not exceed $2 \cdot 10^5$.

Output

For each test case output one integer — the number of tables that you can occupy so as not to violate the rules of the restaurant. If additional tables cannot be taken, then, obviously, you need to output 0.

Sample 1

Input	Output
6 6 1 100010 6 2 000000 5 1 10101 3 1 001 2 2 00 1 1 0	1 2 0 1 1 1

Note

The first test case is explained in the statement.

In the second test case, the answer is 2, since you can choose the first and the sixth table.

In the third test case, you cannot take any free table without violating the rules of the restaurant.

Problem J. Guilty Prince

Time limit	1000 ms
Mem limit	65536 kB

Once there was an Emperor named Akbar. He had a son named Jahangir. For an unforgivable reason, the king wanted him to leave the kingdom. Since he loved his son, he decided his son would be banished to a new place. The prince became sad, but he followed his father's will. On the way, he found that the place was a combination of land and water. Since he didn't know how to swim, he was only able to move on the land. He didn't know how many places might be his destination. So, he asked for your help.

For simplicity, you can consider the place as a rectangular grid consisting of some cells. A cell can be a land or can contain water. Each time the prince can move to a new cell from his current position if they share a side.

Now write a program to find the number of cells (unit land) he could reach including the cell he was initially in.

Input

Input starts with an integer **T (≤ 500)**, denoting the number of test cases.

Each case starts with a line containing two positive integers **W** and **H**; **W** and **H** are the numbers of cells in the **x** and **y** directions, respectively. **W** and **H** will not be more than 20.

There will be **H** more lines in the data set, each of which includes **W** characters. Each character represents the status of a cell as follows.

1. `.` - land.
2. `#` - water.
3. `@` - initial position of the prince (appears exactly once in a dataset).

Output

For each case, print the case number and the number of cells he can reach from the initial position (including self).

Sample

Input	Output

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>

```
4
6 9
....#.
.....#
......
......
......
......
......
#@...#
.#..#.
11 9
.#.........
.#.#######.
.#.#.....#.
.#.#.###.#.
.#.#..@#.#.
.#.#####.#.
.#.......#.
.#########.
...........
11 6
..#..#..#..
..#..#..#..
..#..#..###
..#..#..#@.
..#..#..#..
..#..#..#..
7 7
..#.#..
..#.#..
###.###
...@...
###.###
..#.#..
..#.#..
```

</td><td>

```
Case 1: 45
Case 2: 59
Case 3: 6
Case 4: 13
```

</td></tr>
</table>