Aufgaben
Get Connected to ICPC
Local Contest 2010

Zugang zum Contest:

Wiki: `http://icpc.ira.uka.de/getconnected/contest`
Wettbewerbsserver: `http://domjudge.iti.uni-karlsruhe.de/team/`

Inhaltsverzeichnis

Train Swapping

Description

At an old railway station, you may still encounter one of the last remaining "train swappers". A train swapper is an employee of the railroad, whose sole job it is to rearrange the carriages of trains.

Once the carriages are arranged in the optimal order, all the train driver has to do is drop the carriages off, one by one, at the stations for which the load is meant.

The title "train swapper" stems from the first person who performed this task, at a station close to a railway bridge. Instead of opening up vertically, the bridge rotated around a pillar in the center of the river. After rotating the bridge 90 degrees, boats could pass left or right.

The first train swapper had discovered that the bridge could be operated with at most two carriages on it. By rotating the bridge 180 degrees, the carriages switched place, allowing him to rearrange the carriages. As a side effect, the carriages then faced the opposite direction, but train carriages can move either way.

Now that almost all train swappers have died out, the railway company would like to automate their operation. Part of the program to be developed, is a routine which decides for a given train the least number of swaps of two adjacent carriages necessary to order the train. Your assignment is to create that routine.

Input

The input contains on the first line the number of test cases (N). Each test case consists of two input lines. The first line of a test case contains an integer L, determining the length of the train ($0 \leq L \leq 50$). The second line of a test case contains a permutation of the numbers 1 through L, indicating the current order of the carriages. The carriages should be ordered such that carriage 1 comes first, then 2, etc. with carriage L coming last.

Output

For each test case output a line with the sentence:

```
Optimal train swapping takes S swaps.
```

S is the minimum number of swaps.

Sample Input

```
3
3
1 3 2
4
4 3 2 1
2
2 1
```

Sample Output

```
Optimal train swapping takes 1 swaps.
Optimal train swapping takes 6 swaps.
Optimal train swapping takes 1 swaps.
```

Toll

Description

You are a truck driver that needs to deliver some cargo from one city to another. To accomplish this you are going to use the road network. A road connects exactly 2 cities. Not all cities are directly connected. For example if you want to get from the city A to the city B but the only roads that exist connect A with C and B with C then you will have to pass through the city C. Between 2 cities there is at most one road. Roads may always be used in both direction. Don't make any further assumptions about the topology of the road network. To pass a road you have to pay a tax of x euro. All roads cost the same regardless of their length.

How much do you have to pay at least to complete your trip?

Input

The input begins with a single positive integer T on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

Each test case starts with a line containing 5 positive integers: x, n, m, s and t.

- x is tax amount you have to pay per road ($1 \leq x \leq 1000$)
- n is the number of cities. The cities are number from 1 to n. ($1 \leq n \leq 10000$)
- m is the number of roads. ($1 \leq m \leq 10000$)
- s is the city you start in. ($1 \leq s \leq n$)
- t is the city you want to get to. ($1 \leq t \leq n$)

After that the test case contains m lines each describing a road using its 2 end cities.

Output

The output should contain exactly T lines. The t-th line should contain the the amount you have to pay for the t-th testcase or -1 if you can't reach the city t. A line ends per definition with a newline character.

Sample Input

```
3

3 3 2 1 2
1 3
3 2

100 5 6 4 3
1 3
2 4
5 1
3 2
4 1
2 5

42 6 6 1 6
1 2
2 3
3 1
4 5
5 6
6 4
```

Sample Output

```
6
200
-1
```

Pocket Calculator

Description

The most powerful computer that your friend has ever used was a pocket calculator. Now, that he has a new computer, he is a bit disappointed, because he liked the LC-display of his calculator so much. So you decide to write a program that displays numbers in an LC-display-like style on his computer.

Input

The input file contains several lines, one for each number to be displayed. Each line contains two integers s, n ($1 \leq s \leq 10$, $0 \leq n \leq 99999999$), where n is the number to be displayed and s is the size in which it shall be displayed. No number will begin with a 0 digit except 0 itself.

The input file will be terminated by a line containing two zeros. This line should not be processed.

Output

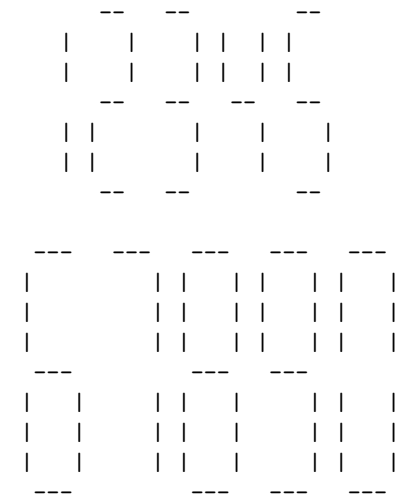
Output the numbers given in the input file in an LC-display-style using s - signs for the horizontal segments and s | signs for the vertical ones. Each digit occupies exactly s+2 columns and 2s+3 rows. (Be sure to fill all the white space occupied by the digits with blanks, also for the last digit.) There has to be exactly one column of blanks between two digits.

Output a blank line after each number. (You will find a sample of each digit in the sample output.)

Sample Input

```
2 12345
3 67890
0 0
```

Sample Output



Joseph

Description

The Joseph's problem is notoriously known. For those who are not familiar with the original problem: From among n people, numbered 1, 2, ..., n, standing in circle every m-th is going to be executed and only the life of the last remaining person will be saved. Joseph was smart enough to choose the position of the last remaining person, thus saving his life to give us the message about the incident. For example when $n = 6$ and $m = 5$ then the people will be executed in the order 5, 4, 6, 2, 3 and 1 will be saved.

Suppose that there are k good guys and k bad guys. In the circle the first k are good guys and the last k bad guys. You have to determine a minimal m such that all the bad guys will be executed before the first good guy.

Input

The input file consists of separate lines containing k. The last line in the input file contains 0. You may assume that $0 < k < 14$.

Output

The output file will consist of separate lines containing m corresponding to k in the input file.

Sample Input

```
3
4
0
```

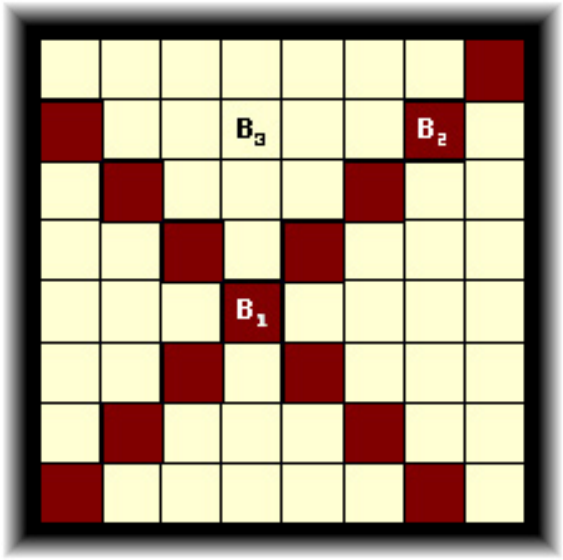
Sample Output

```
5
30
```

Little Bishops

Description

A bishop is a piece used in the game of chess which is played on a board of square grids. A bishop can only move diagonally from its current position and two bishops attack each other if one is on the path of the other. In the following figure, the dark squares represent the reachable locations for bishop B1 form its current position. The figure also shows that the bishops B1 and B2 are in attacking positions whereas B1 and B3 are not. B2 and B3 are also in non-attacking positions.



Now, given two numbers n and k, your job is to determine the number of ways one can put k bishops on an n $\times$ n chessboard so that no two of them are in attacking positions.

Input

The input file may contain multiple test cases. Each test case occupies a single line in the input file and contains two integers n ($1 \leq$ n ≤ 8) and k ($0 \leq$ k $\leq n^2$).

A test case containing two zeros for n and k terminates the input and you won't need to process this particular input.

Output

For each test case in the input print a line containing the total number of ways one can put the given number of bishops on a chessboard of the given size so that no two of them are in attacking positions. You may safely assume that this number will be less than 10^{15} .

Sample Input

```
8 6
4 4
0 0
```

Sample Output

```
5599888
260
```

List Sum

Description

You are given four lists A, B, C and D containing integers. All lists have the same size n. How many quadruplets (a, b, c, d) $\in$ A x B x C x D exist so that a+b+c+d = 0?

Input

The input begins with a single positive integer T on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line of the input file contains the size of the lists n (this value can be as large as 4000). We then have n lines containing four integer values (with absolute value as large as 2^{28}) that belong respectively to A, B, C and D.

Output

For each testcase you should output exactly one line containing the number of possibilities. This means your output should not contain any spaces and exactly T newline characters.

Sample Input

```
2

1
1 2 3 4

6
-45 22 42 -16
-41 -27 56 30
-36 53 -37 77
-36 30 -75 -46
26 -38 -10 62
-32 -54 -6 45
```

Sample Output

```
0
5
```

Sample Explanation: The sum of the five following quadruplets is zero: (-45, -27, 42, 30), (26, 30, -10, -46), (-32, 22, 56, -46),(-32, 30, -75, 77), (-32, -54, 56, 30).