

Thank you for applying to the Software Engineer, University Graduate position at Google. The next step of the application process is an online coding quiz. You will be able to see the coding problems when it starts, but please make sure to read the information below before proceeding.

Date and Time:

- Tuesday, February 25th, 18:00 - 20:00 (Japan Standard Time)

Required tools:

- You will need a computer with access to the internet and development environment (for compiling and executing the programs you write)

Rules of participation:

- The online coding quiz consists of two tasks. When the quiz starts, you will be able to see a menu on the left side containing the coding problems. Feel free to start with the task that you feel more comfortable with and you can use your favorite programming language to solve them.
- The time limit is 120 minutes for the two problems. Remaining time will be shown at the top right section of this site.
- Each task provides input data and directions for generating an output. Your program can process the input in any way that you'd like (e.g. it can read the data from the standard input or a file, or you can embed the data directly in your source code).
- Please submit the output and the source code you wrote using the answer submission section at the bottom of each problem.
- Please do not include personal information in your code and output.
- You can submit the output and the source code as many times as you'd like until the time limit, but only your last correct submission for each task will be considered. (In case you cannot make a correct submission, your last submission will be considered.)
- Even if your code is incomplete it will be considered for partial credit as long as it's submitted within the time limit. If you have no output, please generate a file containing "N/A" and submit it as the output file.
- You must submit your answers within the time limit. The accuracy of your output is the most important factor in your evaluation. The quality of your code, the speed of submission and the number of times you submit are far less important.
- During the quiz you are allowed to browse other websites ONLY to access online dictionaries and to see references and usages of programming languages and libraries.

Important points:

- In case of any issues, notifications will be highlighted in red at the top of this site or you will be notified by email.
- This quiz is to simply assess your coding skills. We will also review and take into consideration your resume and other factors when making a decision on your application.
- All information provided during the recruiting process is confidential. Any unauthorized use, disclosure or distribution of information is prohibited, including, but not limited to, discussion with others and/or posting on websites or any message boards. If any form of misconduct is discovered, it will result in your disqualification.
- This quiz is the first step of the recruiting process. Your technical skills will continue to be evaluated during our recruiting process shall you proceed to the next stages.

If you encounter any problems, please contact us at googlejapan-tech-campus@google.com

Problem description

A ramen shop offers N flavors of ramen along with M options. Flavors determine the taste of broth, such as shōyu (soy sauce), miso (soy bean paste), or tonkotsu (pork marrow). Options can be toppings (such as eggs and pork chops) and side dishes (such as rice bowls and fried chickens).

A customer there plans to spend around X Japanese yen. The customer should order exactly one flavor of ramen, and may order zero, one, or two options. Each option may be ordered only once (e.g. no “two eggs”).

Given the list of available flavors and options, each with its price, what is the price closest to X of possible orders? Here, a price is said closer to X when the difference from X is smaller. Note the customer is allowed to make an order that costs more than X .

Input

Test cases will be provided in the following format, using only ASCII characters. The first line contains one integer, C , which is the number of test cases that will follow. The second line is blank. From third line onwards, multiple test cases separated by a blank line will follow. Each test case has the following format:

```
X
N
FlavorName1 FlavorPrice1
FlavorName2 FlavorPrice2
...
FlavorNameN FlavorPriceN
M
OptionName1 OptionPrice1
OptionName2 OptionPrice2
...
OptionNameM OptionPriceM
```

X is the approximate price the customer plans to spend. N is the number of available flavors. FlavorName_i and FlavorPrice_i are the name and price of the i -th flavor respectively. M is the number of available options. OptionName_j and OptionPrice_j are the name and price of the j -th option respectively.

All tokens in a line are separated by a single space. All prices are given in Japanese yen.

Constraints

- All numbers in the input are integers.
- Number of test cases: $1 \leq C \leq 50$
- Customer's budget: $1 \leq X \leq 10000$
- Number of flavors: $1 \leq N \leq 10$
- Number of options: $0 \leq M \leq 10$
- Price of each flavor: $1 \leq \text{FlavorPrice}_i \leq 10000$
- Price of each option: $1 \leq \text{OptionPrice}_j \leq 10000$
- The total price of all options does not exceed 10000.
- Each name (FlavorName_i , OptionName_j) consists of English alphabets, uppercase (A-Z) and lowercase (a-z), and its length is between 1 and 16, inclusive.
- All names are different within each test case.
- The names are provided only to help you understand the test cases and maybe debug your programs. They do not make any effect to the output.

Output

For each test case, output the result in the following format:

Case #k: Y

where k is the index of the test case, starting from 1, and Y is the price closest to X among the orders the customer can make.

If there are multiple possible prices equally close to X, prefer the lower value. For example, if the customer with (X = 1000) has choices to spend 900 yen and 1100 yen, but not any price in between, your program should output 900 rather than 1100.

Sample input

File: [task1-sample-input.txt](#) (You can download the file using right click -> "Save this link as" or similar)

```
8

1000
3
Shoyu 800
Miso 850
Tonkotsu 900
2
Egg 100
Sprout 150

1000
2
Shoyu 850
Miso 900
2
Rice 200
Chickens 250

1000
2
Superior 1100
Original 900
1
Veggies 200

1000
4
Shoyu 800
Miso 800
Shio 800
Tonkotsu 800
1
Egg 100

1000
4
Shoyu 700
```

Problem description

You are given a tree-shaped undirected graph consisting of N nodes and $N-1$ edges (N is up to 50). For a node X in the tree, let $d(X)$ be the distance (the number of edges) from X to its farthest node. Your task is to compute the minimum value of $d(X)$ for the given tree.

The tree has the following properties:

- It is connected.
- It has no cycles.
- For any pair of distinct nodes X and Y in the tree, there is exactly one path connecting X and Y .

Input

Test cases will be provided in the following multiline text format, using only ASCII characters. The first line contains one integer, C , which is the number of test cases that will follow. The second line is blank. From the third line onwards, the test cases separated by a blank line will follow. Each test case describes a tree and has the following format.

```
N
X1 Y1
X2 Y2
...
XN-1 YN-1
```

N is the number of nodes in the tree. Nodes are numbered 1 through N .

$X_i Y_i$ represents an undirected edge between nodes X_i and Y_i . There are $N - 1$ edges.

Guarantees

- All numbers in the input are integers.
- Number of test cases: $1 \leq C \leq 100$
- Number of nodes: $2 \leq N \leq 50$
- Node IDs: $1 \leq X_i \leq N$, $1 \leq Y_i \leq N$, $X_i \neq Y_i$
- The given graph is always a tree.

Note: You can assume that the input data is valid and satisfies all constraints. Your solution does not need to include error handling code.

Output

For each test case, output the result in the following format:

```
Case #k: R
```

where k is the index of the test case, starting from 1, and R is the minimum of $d(X)$ for the given tree.

All tokens in the output should be separated by a single space.

Sample input

File: [task2-sample-input.txt](#) (You can download the file using right click -> "Save this link as" or similar)

Sample input

File: [task2-sample-input.txt](#) (You can download the file using right click -> "Save this link as" or similar)

```
5

6
1 4
2 3
3 4
4 5
5 6

6
1 3
4 5
5 6
3 2
3 4

2
1 2

10
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10

10
7 8
7 9
4 5
1 3
3 4
6 7
4 6
2 3
9 10
```

Output for sample input

File: [task2-sample-output.txt](#) (You can download the file using right click -> "Save this link as" or similar)

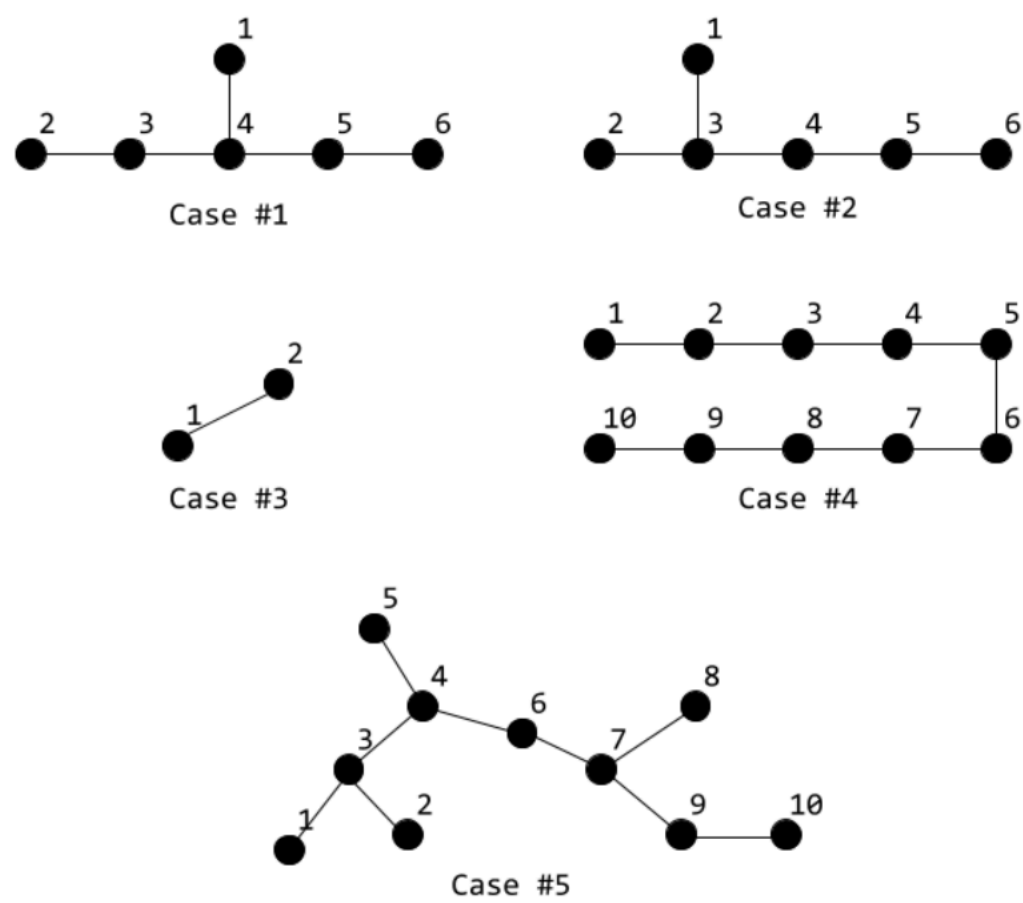
```
Case #1: 2
Case #2: 2
Case #3: 1
Case #4: 5
Case #5: 3
```

Output for sample input

File: [task2-sample-output.txt](#) (You can download the file using right click -> "Save this link as" or similar)

Case #1: 2
Case #2: 2
Case #3: 1
Case #4: 5
Case #5: 3

Visualization of Sample Input



Test input

Please submit output for this input in the submission form below.

File: [task2-test-input.txt](#) (You can download the file using right click -> "Save this link as" or similar)

Submit

Source Code:

no file selected