

Round B APAC Test 2017

A. Sherlock and Parentheses

[B. Sherlock and Watson Gym Secrets](#)

[C. Watson and Intervals](#)

[D. Sherlock and Permutation Sorting](#)

Questions asked 1

Submissions

Sherlock and Parentheses

4pt	Not attempted 3846/5689 users correct (68%)
7pt	Not attempted 2912/3801 users correct (77%)

Sherlock and Watson Gym Secrets

6pt	Not attempted 1760/3710 users correct (47%)
15pt	Not attempted 268/1026 users correct (26%)

Watson and Intervals

8pt	Not attempted 526/1376 users correct (38%)
17pt	Not attempted 152/284 users correct (54%)

Sherlock and Permutation Sorting

19pt	Not attempted 44/428 users correct (10%)
24pt	Not attempted 15/27 users correct (56%)

Top Scores

bsbandme	100
alecsyde	100
RiverBlessPeople	100
NAFIS	100
izrak	100
dragon7	100
winoros	100
gvaibhav21	100
stonebuddha	100
VastoLorde95	100

Problem A. Sherlock and Parentheses

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
4 points

Solve A-small

Large input
7 points

Solve A-large

Problem

Sherlock and Watson have recently enrolled in a computer programming course. Today, the tutor taught them about the balanced parentheses problem. A string S consisting only of characters $($ and/or $)$ is *balanced* if:

- It is the empty string, or:
- It has the form (S) , where S is a balanced string, or:
- It has the form S_1S_2 , where S_1 is a balanced string and S_2 is a balanced string.

Sherlock coded up the solution very quickly and started bragging about how good he is, so Watson gave him a problem to test his knowledge. He asked Sherlock to generate a string S of $\mathbf{L} + \mathbf{R}$ characters, in which there are a total of \mathbf{L} left parentheses $($ and a total of \mathbf{R} right parentheses $)$. Moreover, the string must have as many different balanced non-empty substrings as possible. (Two substrings are considered different as long as they start or end at different indexes of the string, even if their content happens to be the same). Note that S itself does not have to be balanced.

Sherlock is sure that once he knows the maximum possible number of balanced non-empty substrings, he will be able to solve the problem. Can you help him find that maximum number?

Input

The first line of the input gives the number of test cases, \mathbf{T} . \mathbf{T} test cases follow. Each test case consists of one line with two integers: \mathbf{L} and \mathbf{R} .

Output

For each test case, output one line containing Case # x : y , where x is the test case number (starting from 1) and y is the answer, as described above.

Limits

$1 \leq \mathbf{T} \leq 100$.

Small dataset

$0 \leq \mathbf{L} \leq 20$.
 $0 \leq \mathbf{R} \leq 20$.
 $1 \leq \mathbf{L} + \mathbf{R} \leq 20$.

Large dataset

$0 \leq \mathbf{L} \leq 10^5$.
 $0 \leq \mathbf{R} \leq 10^5$.
 $1 \leq \mathbf{L} + \mathbf{R} \leq 10^5$.

Sample

Input	Output
3	Case #1: 0
1 0	Case #2: 1
1 1	Case #3: 3
3 2	

In Case 1, the only possible string is $()$. There are no balanced non-empty substrings.

In Case 2, the optimal string is $()()$. There is only one balanced non-empty substring: the entire string itself.

In Case 3, both strings $()()()$ and $((()))$ give the same optimal answer.

For the case $()()()$, for example, the three balanced substrings are $()$ from indexes 1 to 2, $()$ from indexes 3 to 4, and $()()$ from indexes 1 to 4.

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform