
Unrank pairs of parentheses

P20584_en

In general, there are many ways to place n pairs of parentheses correctly. For instance, these are just a few of the 42 ways for $n = 5$:

`()()()()()` `()(())(())` `((())(())` `(((()(())` `(((((())`)

The following rules inductively define all the correct strings made up with parentheses:

- The empty string is correct.
- All correct non-empty strings are of the kind $(x)y$, where x and y are correct strings.

Let $|s|$ denote the length of a string s . We can define as follows a total order among the correct strings with parentheses:

- The empty string is smaller than any non-empty string.
- Given two non-empty strings $s_1 = (x_1)y_1$ and $s_2 = (x_2)y_2$, s_1 is smaller than s_2 if and only if:
 - $|s_1| < |s_2|$,
 - or $|s_1| = |s_2|$ and x_1 is smaller than x_2 ,
 - or $|s_1| = |s_2|$, $x_1 = x_2$ and y_1 is smaller than y_2 .

Can you write a program to compute the i -th correct string with n pairs of parentheses?

Input

Input consists of several cases, each one with two numbers i and n . Assume $0 \leq n \leq 30$ and that i is between 1 and the number of correct strings with n pairs of parentheses.

Output

For every case, print the i -th correct string with n pairs of parentheses.

Sample input

```
1 3
2 3
3 3
4 3
5 3
70 6
1 0
1 30
200000000000000000 30
3814986502092304 30
```

Sample output

```
()()()
()((()
((()()
((()()
(((())
((()((()((()

()()()()()()()()()()()()()()()()()()()()()()()()()()()()
(((())()((()((()((()((()((()((()((()((()((()((()((()((()((()
((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

Problem information

Author : Salvador Roura
Generation : 2013-09-20 14:46:10

© *Jutge.org*, 2006–2013.
<http://www.jutge.org>