

110604 Expressions

Let X be the set of *correctly built parenthesis expressions*. The elements of X are strings consisting only of the characters “(” and “)”, defined as follows:

- The empty string belongs to X .
- If A belongs to X , then (A) belongs to X .
- If both A and B belong to X , then the concatenation AB belongs to X .

For example, the strings $()(())()$ and $((()()))$ are correctly built parenthesis expressions, and therefore belong to the set X . The expressions $((()))(())$ and $()(())$ are not correctly built parenthesis expressions and are thus not in X .

The *length* of a correctly built parenthesis expression E is the number of single parenthesis (characters) in E . The *depth* $D(E)$ of E is defined as follows:

$$D(E) = \begin{cases} 0 & \text{if } E \text{ is empty} \\ D(A) + 1 & \text{if } E = (A), \text{ and } A \text{ is in } X \\ \max(D(A), D(B)) & \text{if } E = AB, \text{ and } A, B \text{ are in } X \end{cases}$$

For example, $()(())()$ has length 8 and depth 2. Write a program which reads in n and d and computes the number of correctly built parenthesis expressions of length n and depth d .

Input

The input consists of pairs of integers n and d , with at most one pair per line and $2 \leq n \leq 300$, $1 \leq d \leq 150$. The input may contain empty lines, which you don't need to consider.

Output

For every pair of integers in the input, output a single integer on one line – the number of correctly built parenthesis expressions of length n and depth d .

Sample Input

```
6 2
300 150
```

Sample Output

```
3
1
```

Note: The three correctly built parenthesis expressions of length 6 and depth 2 are $((()))()$, $()(())$, and $(())()$.