ALGORITHM DESIGN TERM PROJECT

1222. CHERNOBYL' EAGLES

Grid Kornsutatipkul 6013732

CS3201 – Algorithm Design

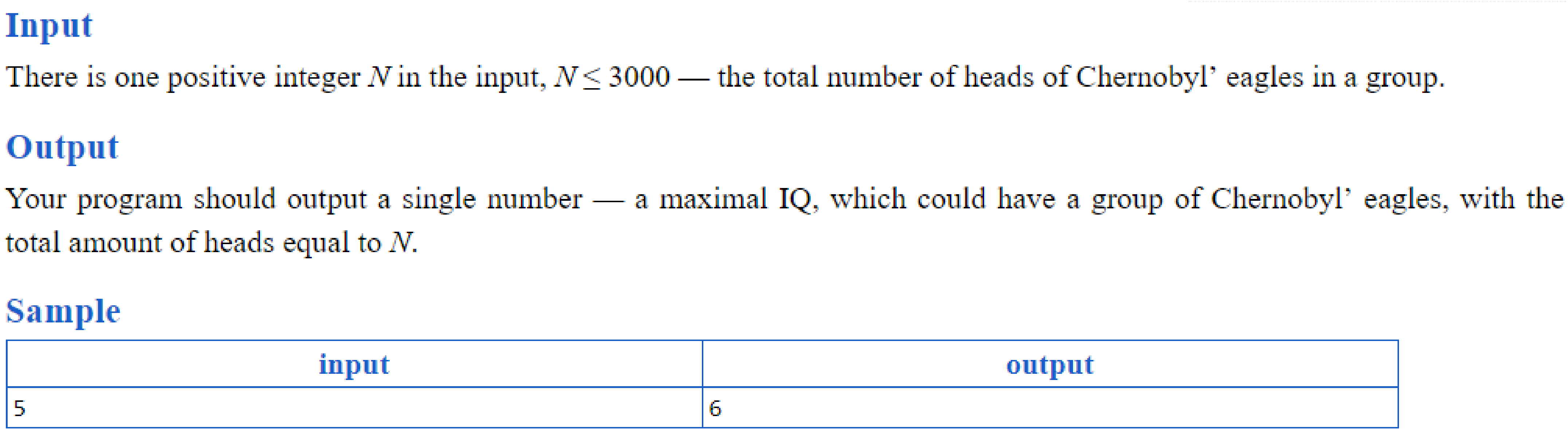
Introduction

Time limit: 1.0 second
Memory limit: 64 MB
Difficulty: 140

Description

A Chernobyl' eagle has several heads (for example, the eagle on the Russian National Emblem is a very typical one, having two heads; there exist Chernobyl' eagles having twenty-six, one and even zero heads). As all eagles, Chernobyl' eagles are very intelligent. Moreover, IQ of a Chernobyl' eagle is exactly equal to the number of its heads. These eagles can also enormously enlarge their IQ, when they form a group for a brainstorm. IQ of a group of Chernobyl' eagles equals to the product of IQ's of eagles in the group. So for example, the IQ of a group, consisting of two 4-headed eagles and one 7-headed is 4*4*7=112. The question is, how large can be an IQ of a group of eagles with a given total amount of heads.

Problem



Input

There is one positive integer N in the input, $N \leq 3000$ — the total number of heads of Chernobyl' eagles in a group.

Output

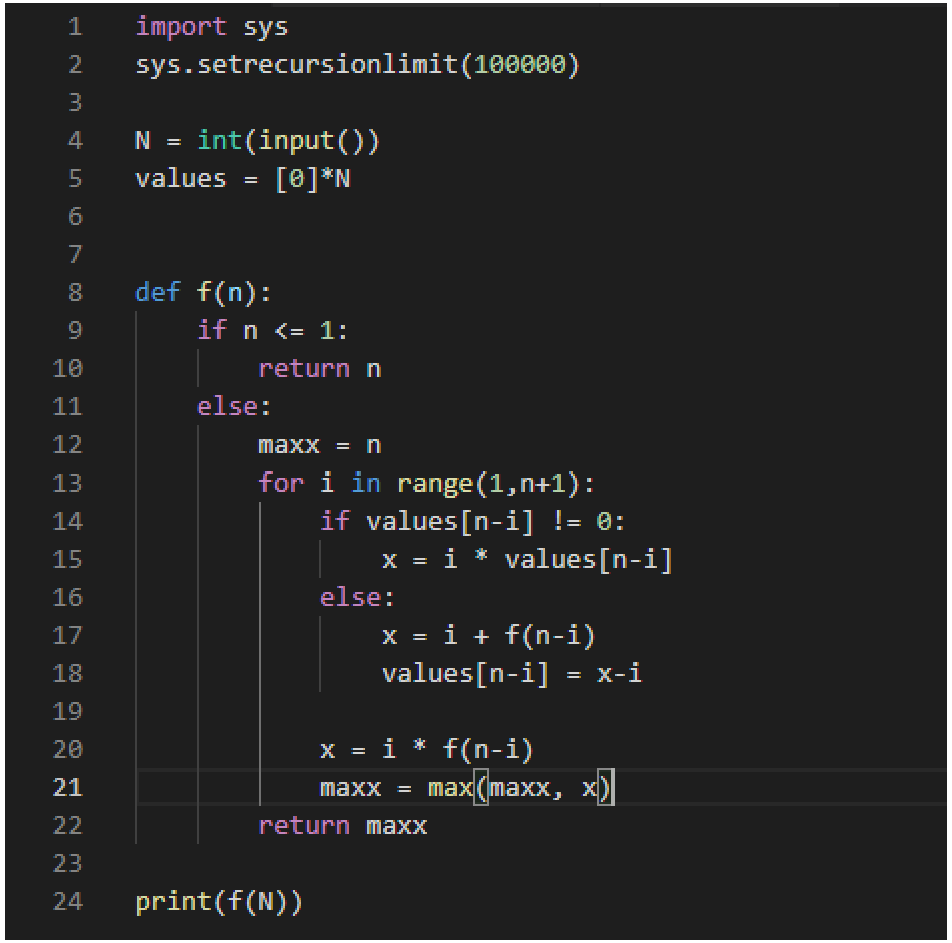
Your program should output a single number — a maximal IQ, which could have a group of Chernobyl' eagles, with the total amount of heads equal to N.

Sample

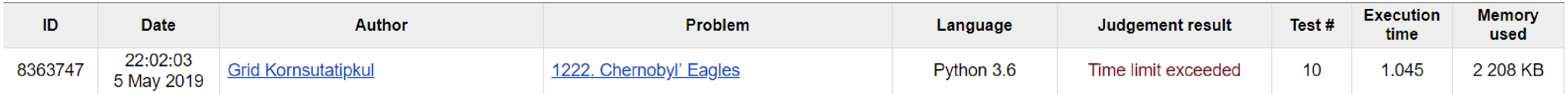
input	output
5	6

Solution

First version applied speed up cutting rod algorithm



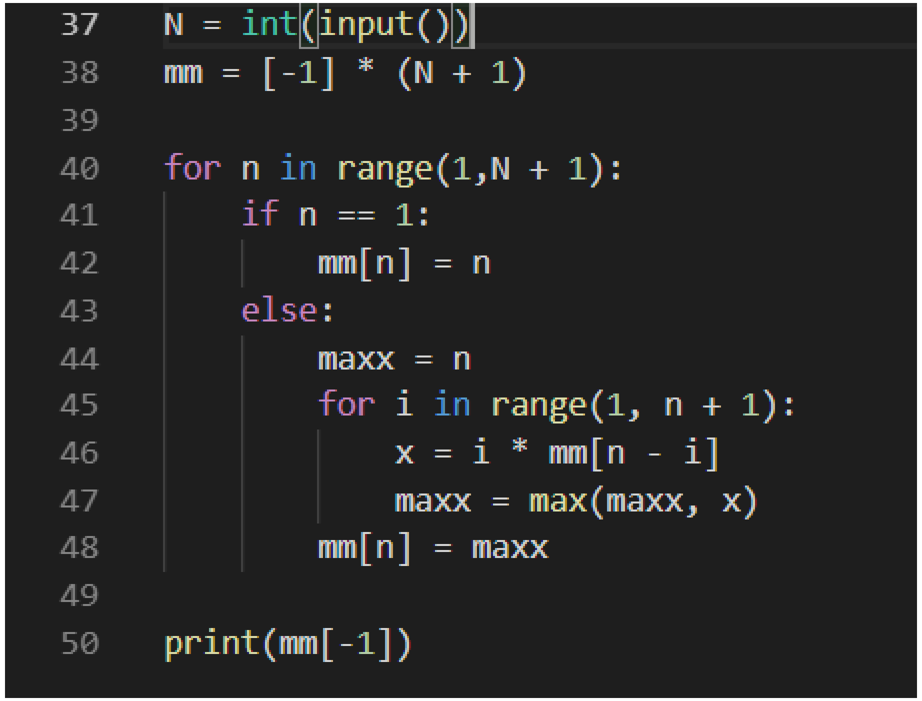
```python
import sys
sys.setrecursionlimit(100000)

N = int(input())
values = [0]*N


def f(n):
    if n <= 1:
        return n
    else:
        maxx = n
        for i in range(1,n+1):
            if values[n-i] != 0:
                x = i * values[n-i]
            else:
                x = i + f(n-i)
                values[n-i] = x-i

            x = i * f(n-i)
            maxx = max(maxx, x)
        return maxx

print(f(N))
```



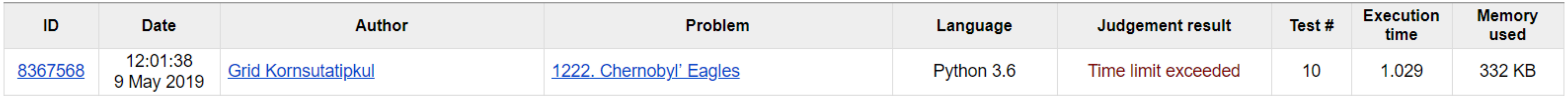
ID	Date	Author	Problem	Language	Judgement result	Test #	Execution time	Memory used
8363747	22:02:03 5 May 2019	Grid Kornsutatipkul	1222. Chernobyl' Eagles	Python 3.6	Time limit exceeded	10	1.045	2 208 KB

Solution

Dynamic programming cutting rod algorithm

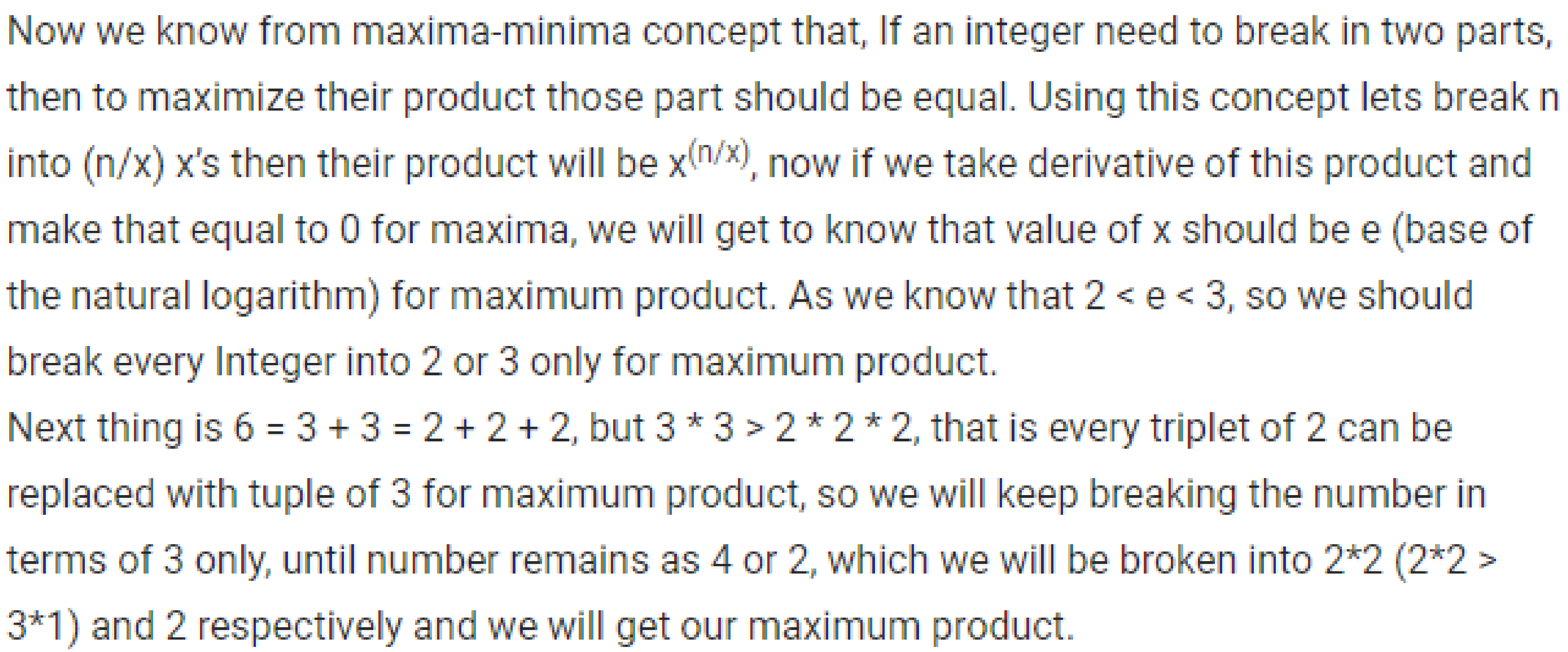


```python
N = int(input())
mm = [-1] * (N + 1)

for n in range(1,N + 1):
    if n == 1:
        mm[n] = n
    else:
        maxx = n
        for i in range(1, n + 1):
            x = i * mm[n - i]
            maxx = max(maxx, x)
        mm[n] = maxx

print(mm[-1])
```



ID	Date	Author	Problem	Language	Judgement result	Test #	Execution time	Memory used
8367568	12:01:38 9 May 2019	Grid Kornsutatipkul	1222. Chernobyl' Eagles	Python 3.6	Time limit exceeded	10	1.029	332 KB

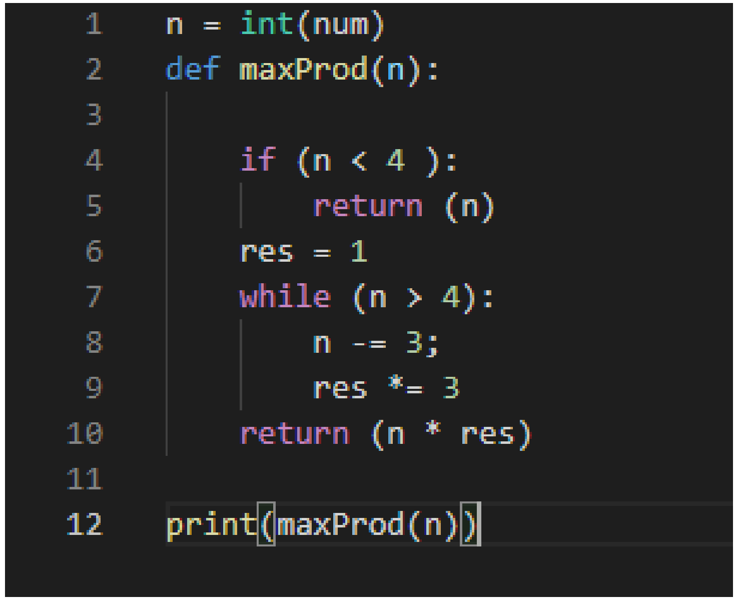
Solution



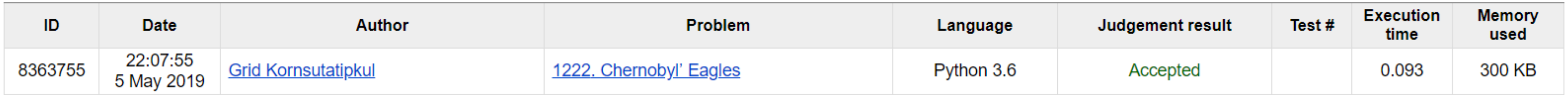
Now we know from maxima-minima concept that, If an integer need to break in two parts, then to maximize their product those part should be equal. Using this concept lets break n into (n/x) x's then their product will be $x^{(n/x)}$, now if we take derivative of this product and make that equal to 0 for maxima, we will get to know that value of x should be e (base of the natural logarithm) for maximum product. As we know that $2 < e < 3$, so we should break every Integer into 2 or 3 only for maximum product.
Next thing is 6 = 3 + 3 = 2 + 2 + 2, but 3 * 3 > 2 * 2 * 2, that is every triplet of 2 can be replaced with tuple of 3 for maximum product, so we will keep breaking the number in terms of 3 only, until number remains as 4 or 2, which we will be broken into 2*2 (2*2 > 3*1) and 2 respectively and we will get our maximum product.

Solution

Find maximum product with cutting part of size 3



```python
n = int(num)
def maxProd(n):

    if (n < 4 ):
        return (n)
    res = 1
    while (n > 4):
        n -= 3;
        res *= 3
    return (n * res)

print(maxProd(n))
```



ID	Date	Author	Problem	Language	Judgement result	Test #	Execution time	Memory used
8363755	22:07:55 5 May 2019	Grid Kornsutatipkul	1222. Chernobyl' Eagles	Python 3.6	Accepted		0.093	300 KB

References

https://www.geeksforgeeks.org/breaking-integer-to-get-maximum-product/
https://www.geeksforgeeks.org/maximum-product-cutting-dp-36/