

Test Name:

- Summary
- Timeline

Tasks summary

Task	Time spent	Score
ChocolatesByNumbers Python	2 min	100%

Total score

100%

Tasks Details

Easy

1. ChocolatesByNumbers

There are N chocolates in a circle. Count the number of chocolates you will eat.

Task Score

100%

Correctness

100%

Performance

100%

Task description

Two positive integers N and M are given. Integer N represents the number of chocolates arranged in a circle, numbered from 0 to N – 1.

You start to eat the chocolates. After eating a chocolate you leave only a wrapper.

You begin with eating chocolate number 0. Then you omit the next M – 1 chocolates or wrappers on the circle, and eat the following one.

More precisely, if you ate chocolate number X, then you will next eat the chocolate with number (X + M) modulo N (remainder of division).

You stop eating when you encounter an empty wrapper.

For example, given integers N = 10 and M = 4. You will eat the following chocolates: 0, 4, 8, 2, 6.

The goal is to count the number of chocolates that you will eat, following the above rules.

Write a function:

```
def solution(N, M)
```

that, given two positive integers N and M, returns the number of chocolates that you will eat.

For example, given integers N = 10 and M = 4. the function should return 5, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N and M are integers within the range [1..1,000,000,000].

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

Programming language used: Python

Total time used: 2 minutes

?

Effective time used: 2 minutes

?

Notes:

not defined yet

Task timeline

21:01:3621:03:30

Code: 21:03:30 UTC, py, final, show code in pop-up

score: 100

```
1  # you can write to stdout for debugging purposes,
2  # print("this is a debug message")
3
4  def solution(N, M):
5      # write your code in Python 3.6
6      if N == M:
7          return 1
8      if N%M == 0:
9          return N//M
10     if M < N:
11         i = 2
12     else:
13         i = M//N
14     while True:
15         if M%i == 0:
16             if (i*N)%M == 0:
17                 return (i*N)//M
18             i = i + 1
19             if i > M//2:
20                 break
21     return N
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(log(N + M))

expand all

Example tests

▶ example

example test

OK

expand all

Correctness tests

▶ extreme_small

very small N and M

OK

▶ simple

simple test, N = 24, M = 18

OK

▶ small1

small tests

OK

▶ small2

small tests

OK

expand all

Performance tests

▶ medium

medium tests

OK

▶ large

large tests

OK

▶ large2

N = (3**9)*(2**14), M=(2**14)*(2**14)

OK

▶ extreme_large

maximal and minimal values

OK