

Яндекс. Тренировки по алгоритмам июнь 2021, занятие 6

Г. Площадь

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.5 секунд	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
GNU C11 7.3	0.3 секунды	256Mb		
GNU c++ 11 4.9	0.3 секунды	256Mb		
GNU c++ 11 x32 4.9	0.3 секунды	256Mb		
GCC 5.4.0 C++14	0.3 секунды	256Mb		
GNU c++ 14 4.9	0.3 секунды	256Mb		
GNU c++17 7.3	0.3 секунды	256Mb		
GCC C++17	0.3 секунды	256Mb		

Городская площадь имеет размер $n \times m$ и покрыта квадратной плиткой размером $l \times l$. При плановой замене плитки выяснилось, что новой плитки недостаточно для покрытия всей площади, поэтому было решено покрыть плиткой только дорожку по краю площади, а в центре площади разбить прямоугольную клумбу (см. рисунок к примеру). При этом дорожка должна иметь одинаковую ширину по всем сторонам площади. Определите максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Формат ввода

Первая и вторая строки входных данных содержат по одному числу n и m ($3 \leq n \leq 2 \times 10^9$, $3 \leq m \leq 2 \times 10^9$) — размеры площади. Третья строка содержит количество имеющихся плиток t , $1 \leq t < nm$.

Обратите внимание, что значение t может быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные числа (тип `int64` в языке Pascal, тип `long long` в C и C++, тип `long` в Java и C#).

Формат вывода

Программа должна вывести единственное число — максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Пример

Ввод

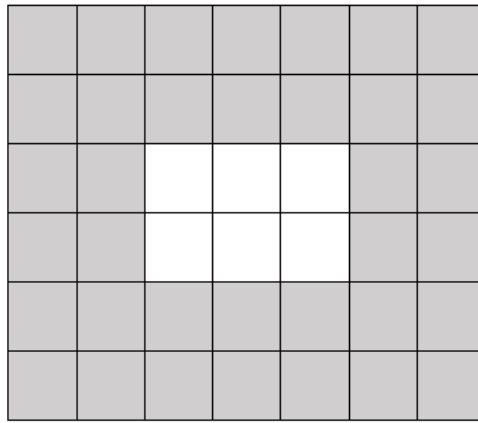
Вывод

6
7
38

2

Примечания

Пояснение к примеру. Площадь имеет размеры 6×7 , из 38 плиток можно выложить дорожку шириной в 2 плитки.



Две плитки
осталось



Язык Python 3.12.1

Набрать здесь

Отправить файл

```
1 n = int(input().strip())
2 m = int(input().strip())
3 t = int(input().strip())
4
5
6 lf, rg = 0, max(n//2, m//2)
7 # левый бин поиск
8 while lf < rg:
9     # считаем центр
10    mid = (lf + rg) // 2 + 1
11    # если не успели напечатать N - 1 копий - сдвигаем lf на mid + 1
12    count = (mid * 2 * n) + ((m - 2 * mid) * 2 * mid)
13    if count <= t:
14        lf = mid
15    # иначе сдвигаем rg на mid
16    else:
17        rg = mid - 1
18
19 # ответ
20 print(lf)
```

Отправить

Предыдущая

Следующая