

Яндекс. Тренировки по алгоритмам июнь 2021, занятие 6

D. Космическое поселение

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Для освоения Марса требуется построить исследовательскую базу. База должна состоять из n одинаковых модулей, каждый из которых представляет собой прямоугольник.

Каждый модуль представляет собой жилой отсек, который имеет форму прямоугольника размером a на b метров. Для повышения надежности модулей инженеры могут добавить вокруг каждого модуля слой дополнительной защиты. Толщина этого слоя должна составлять целое число метров, и все модули должны иметь одинаковую толщину дополнительной защиты. Модуль с защитой, толщина которой равна d метрам, будет иметь форму прямоугольника размером $(a+2d)(b+2d)$ метров.

Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером w на h метров. При этом они должны быть организованы в виде регулярной сетки: их стороны должны быть параллельны сторонам поля, и модули должны быть ориентированы одинаково.

Требуется написать программу, которая по заданному количеству и размеру модулей, а также размеру поля для их размещения, определяет максимальную толщину слоя дополнительной защиты, который можно добавить к каждому модулю.

Формат ввода

Входной файл содержит пять разделенных пробелами целых чисел: n , a , b , w и h ($1 \leq n, a, b, w, h \leq 10^4$). Гарантируется, что без дополнительной защиты все модули можно разместить в поселении описанным образом.

Формат вывода

Выходной файл должен содержать одно целое число: максимальную возможную толщину дополнительной защиты. Если дополнительную защиту установить не удастся, требуется вывести число 0.

Пример 1

Ввод

1 1 1 1 1

Вывод

0

Пример 2

Ввод

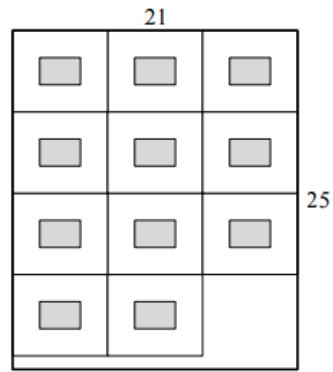
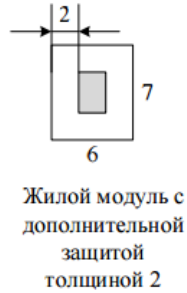
1 1 1 3 3

Вывод

1

Примечания

Для входных данных 11 3 2 21 25 можно установить дополнительную защиту толщиной 2 метра и разместить модули на поле, как показано на рисунке.



Язык Python 3.12.1

Набрать здесь Отправить файл

```
1 from math import ceil
2
3 # считываем данные
4 n, a, b, w, h = map(int, input().split())
5
6 # используем ПРАВЫЙ бин поиск для первой расстановки
7 lf, rg = 0, (max(w, h) - min(a, b)) // 2
8 while lf < rg:
9     # считаем центр
10    mid = (lf + rg) // 2 + 1
11    # считаем строки и столбцы домов
12    rows = h // (a+2*mid)
13    if rows == 0:
14        rg = mid - 1
15    else:
16        cols = ceil(n/rows)
17        if cols > 0 and cols * (b+2*mid) <= w:
18            lf = mid
19        # если дома не поместились, сдвигаем правый указатель на mid - 1
20        else:
21            rg = mid - 1
22
23 lf_forward = lf
24
25 # используем ПРАВЫЙ бин поиск для расстановки, повернув дома на 90
26 lf, rg = 0, (max(w, h) - min(a, b)) // 2
27 while lf < rg:
28     # считаем центр
29     mid = (lf + rg) // 2 + 1
30     # считаем строки и столбцы домов
31     rows = h // (b+2*mid)
32     if rows == 0:
33         rg = mid - 1
34     else:
35         cols = ceil(n/rows)
36         if cols > 0 and cols * (a+2*mid) <= w:
37             lf = mid
38         # если дома не поместились, сдвигаем правый указатель на mid - 1
```

Отправить

Предыдущая

Следующая