

Разборы задач №1

НОК, математика

Содержание

3 – Задача наНОК – codeforces 235A

9 – Театральная площадь – codeforces 1A

13 – Пицца, пицца, пицца!!! – codeforces 979A

17 – Прыгающая лягушка – codeforces 1077A

23 – Уравнение – informatics №260

Задача на НОК – codeforces 235A

Несколько дней назад я узнал, что существует такая штука как наименьшее общее кратное (НОК). Теперь я часто играю с этим понятием — хочу сделать большое число с помощью НОК.

Но я не хочу использовать слишком много чисел, поэтому я выберу три целых положительных числа (необязательно различных), каждое из которых не превышает n . Помогите мне найти максимально возможное наименьшее общее кратное этих трех целых чисел.

Входные данные: целое число n ($1 \leq n \leq 10^6$) — переменная n из условия.

Выходные данные: единственное целое число — максимально возможное НОК трех обязательно различных целых чисел, которые не превышают n .

Задача на НОК – codeforces 235A

Примеры

Входные данные	Выходные данные
9	504
7	210

Задача на НОК – codeforces 235A

Идея решения для данной задачи основана на проверке кратности N .

В зависимости от того, НОК скольких чисел K мы ищем, нам следует проверить кратность N всем простым числам, не превосходящим K .

Кроме того, очевидно, что результатом в данной задаче будет НОК трех чисел, максимально близких к N . Тогда рассмотрим N .

Задача на НОК – codeforces 235A

Если $N < 3$, то достаточно вернуть N в качестве результата.

Если $N > 3$, то рассмотрим случаи, когда N - нечетное и когда N - четное.

Если N – нечетное, то в качестве результата получится число $N(N-1)(N-2)$.

Если N – четное, то рассмотрим два случая:

1. N делится на 3. Тогда результат – $(N-1)(N-2)(N-3)$.
2. N не делится на 3. Тогда результат – $N(N-1)(N-3)$.

Задача на НОК – codeforces 235A

Примеры:

1. 7 – нечетное, результат – $7*6*5=504$
2. 18 – четное, делится на 3, результат – $18*17*15=4590$
3. 22 – четное, не делится на 3, результат – $22*21*19=8778$

Задача на НОК – codeforces 235A

```
#include <iostream>
using namespace std;
int main() {
    long long n;
    cin>>n;
    if (n<3)
        cout<<n;
    else if (n%2!=0)
        cout<< n*(n-1)*(n-2);
    else if (n%3!=0)
        cout<< n*(n-1)*(n-3);
    else
        cout<<(n-1)*(n-2)*(n-3);
    return 0;
}
```


Театральная площадь – codeforces 1A

Театральная площадь в столице Берляндии представляет собой прямоугольник $n \times m$ метров. По случаю очередного юбилея города, было принято решение о замощении площади квадратными гранитными плитами. Каждая плита имеет размер $a \times a$.

Какое наименьшее количество плит понадобится для замощения площади? Разрешено покрыть плитами большую поверхность, чем театральная площадь, но она должна быть покрыта обязательно. Гранитные плиты нельзя ломать или дробить, а разрешено использовать только целиком. Границы плит должны быть параллельны границам площади.

Входные данные: 3 целых натуральных числа n, m, a ($1 \leq n, m, a \leq 10^9$).

Выходные данные: искомое количество плит

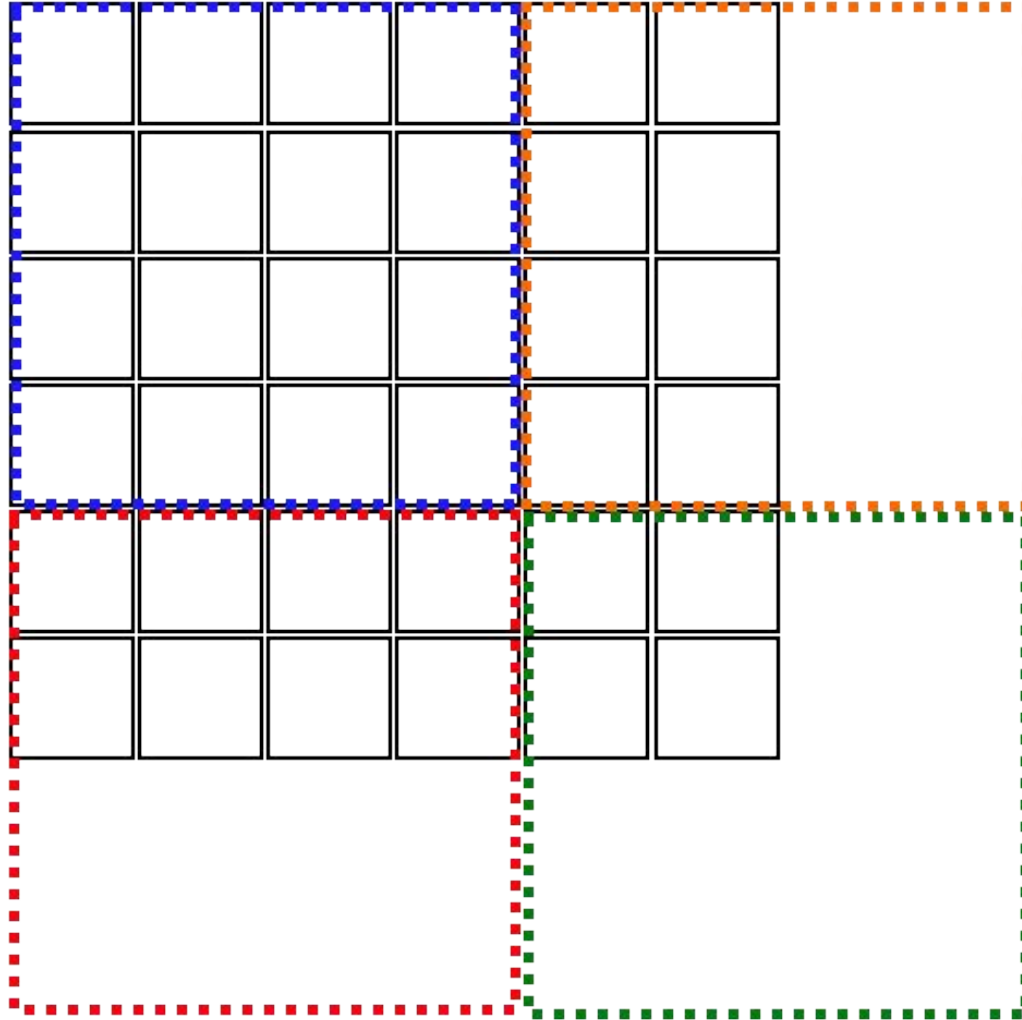
Театральная площадь – codeforces 1A

Примеры

Входные данные	Выходные данные
6 6 4	4

Поскольку сказано, что площадь имеет прямоугольную форму, а также то, что при покрытии площади можно выходить за её границы, нам достаточно умножить наименьшее целое число плит, которые покроют длину площади на наименьшее целое число плит, которые покроют её ширину.

Театральная площадь – codeforces 1A



Театральная площадь – codeforces 1A

```
#include <iostream>
using namespace std;
int main()
{
    long long m, n, a;
    cin>>m>>n>>a;
    cout<<((m+a-1)/a)*((n+a-1)/a);
}
```

Пицца, пицца, пицца!!! – codeforces 979A

Кэти, Куро и Широ — лучшие друзья. Они знают друг друга ещё с детского сада, поэтому очень часто делятся всем между собой и решают сложные задачи вместе.

Сегодня день рождения Широ. Она очень любит пиццу, поэтому она хочет пригласить своих друзей, включая Куро и Кэти, в пиццерию рядом с её домом. Для того, чтобы накормить всех своих друзей, Широ заказала очень большую круглую пиццу. На день рождения придёт ровно n друзей Широ, поэтому ей надо разделить пиццу на $n+1$ кусок (Широ тоже надо есть!) Она хочет, чтобы все куски пиццы были одной и той же формы и одного и того же размера, иначе её друзья будут в бешенстве и испортят всё веселье.

Широ уже голодна, поэтому она хочет разрезать пиццу минимальным числом прямых разрезов, но при этом она слишком ленива, чтобы достать калькулятор. Разрез — это прямой отрезок, он может начинаться и заканчиваться как внутри, так и снаружи пиццы.

Обычно Широ попросила бы помощи у Кэти и Куро, но они ещё не пришли. Можете ли вы помочь Широ решить эту задачу?

Пицца, пицца, пицца!!! – codeforces 979A

Входные данные: целое число n ($0 \leq n \leq 10^{18}$) — число друзей Широ. Пиццу в форме круга надо разрезать на $n+1$ часть.

Выходные данные: одно целое число — количество прямых разрезов, которые необходимо провести Широ.

Примеры

Входные данные	Выходные данные
3	2
4	5

Пицца, пицца, пицца!!! – codeforces 979A

Достаточно знать, что если $N+1$ (число друзей Ширы и она сама) – нечетное, то нам понадобится $N+1$ разрезов, а если четное – $N/2$.

Также следует учитывать, что если $N+1=1$, то пиццу разрезать не нужно.

Пицца, пицца, пицца!!! – codeforces 979A

```
#include <iostream>
using namespace std;
int main(){
    long long n;
    cin>>n;
    n++;
    if (n == 1)
        cout<<0<<endl;
    else if (n % 2 == 0)
        cout<<n/2<<endl;
    else cout<<n;
    return 0;
}
```


Прыгающая лягушка – codeforces 1077A

Сейчас лягушка стоит в позиции 0 на координатной оси Ox . Она прыгает по следующему алгоритму: первый прыжок — на a вправо, второй прыжок — на b влево, третий прыжок — на a вправо, четвёртый прыжок — на b влево, и так далее.

Формально:

если лягушка уже прыгнула четное число раз (перед текущим прыжком), то она прыгает от ее текущей позиции x в позицию $x+a$;

иначе она прыгает от ее текущей позиции x в позицию $x-b$.

Ваша задача — найти позицию лягушки после k прыжков.

Но... Кое-что еще. Вы наблюдаете за t различными лягушками, так что вам нужно ответить на t независимых запросов.

Прыгающая лягушка – codeforces 1077A

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 1000$) — количество запросов.

Каждая из следующих t строк содержит запросы (один запрос на строку).

Запрос описывается в виде трех целых чисел a, b, k ($1 \leq a, b, k \leq 10^9$) — длины прыжков двух типов и количество прыжков соответственно.

Выходные данные

Выведите t целых чисел. i -е число должно быть равно ответу на i -й запрос.

Прыгающая лягушка – codeforces 1077A

Примеры

Входные данные	Выходные данные
6	8
5 2 3	198
100 1 4	-17
1 10 5	29999999997
1000000000 1 6	0
1 1 1000000000	1
1 1 9999999999	

Прыгающая лягушка – codeforces 1077A

- Для решения данной задачи достаточно посчитать итоговое расстояние, которое лягушка преодолеет, прыгнув в каждую сторону. Их разница будет являться одним из ответов.

Прыгающая лягушка – codeforces 1077A

```
#include <iostream>
using namespace std;
int main(){
    long long n, a, b, c;
    cin>>n;
    long long *k=new long long [n];
    for (int i=0;i<n;i++)
    {
        cin>>a>>b>>c;
        k[i]=0-((c/2)*b)+(a*(c-(c/2)));
    }
    for (int i=0;i<n;i++)
        cout<<k[i]<<endl;
    return 0;
}
```

Уравнение – informatics №260

Решить в целых числах уравнение $ax + b = 0$.

Входные данные: 2 целых числа: a и b .

Выходные данные: все решения, если их число конечно, “*NO*” (без кавычек), если решений нет, и “*INF*” (без кавычек), если решений бесконечно много.

Примеры

Входные данные	Выходные данные
6 -2	NO
1 1	-1

Уравнение – informatics №260

Дано уравнение $ax+b=0$.

Если a и b оба равны 0 , то мы получим бесконечное количество решений.

Если a равно нулю, а b - нет, то решений нет т. к. такая комбинация приводит к делению на ноль.

Если b делится на a без остатка, то достаточно вывести a , в противном случае для уравнения нет решения в целых числах.

Уравнение – informatics №260

```
#include <iostream>
using namespace std;
int main() {
    int a, b, x;
    cin>>a>>b;
    if ((a==0)&&(b==0))
        cout<<"INF";
    else if ((a==0) || (b%a!=0))
        cout<<"NO";
    else {
        b=0-b;
        b=b/a;
        cout<<b;}
    return 0;}
```