

## Заказ в магазине

Сначала проверим, существует ли решение. Максимальное количество ручек, которое можно заказать, равно  $1 + 2 + 3 + \dots + N = N(N + 1)/2$  (по формуле суммы арифметической прогрессии), и если  $M > N(N + 1)/2$ , то решения не существует.

При проверке этого условия можно столкнуться с проблемой переполнения 32-битного целого типа в языках C++, Pascal, Java, C#, поэтому вычисление нужно производить с использованием 64-битных целочисленных переменных. Также если эту сумму вычислять не по формуле, а при помощи цикла, то цикл длины  $N$  может не уложиться в ограничение по времени. В этом случае стоит прервать цикл, если сумма превысит  $M$ , либо заметить, что если  $N > \sqrt{2 \cdot 10^9} \approx 44721$ , то  $1 + 2 + 3 + \dots + N > 10^9$  и решение существует.

Если решение существует, то воспользуемся жадным алгоритмом: будем выбирать упаковки максимально возможного размера:  $N$ ,  $N - 1$ ,  $N - 2$  и т.д. Если размер рассматриваемой упаковки  $s$  больше или равен  $M$ , то выведем значение  $s$  и уменьшим  $M$  на  $s$ .

Если использовать цикл `for` от  $N$  до 1, то получится решение сложности  $O(N)$ , которое наберёт 40 баллов. Надо заметить, что вовсе необязательно перебирать все значения от  $N$  до 1, т.к. если в какой-то момент значение  $M$  станет меньше рассматриваемого размера упаковки, то достаточно взять одну упаковку размером  $M$ , то есть вывести  $M$  и завершить работу программы. Такое решение будет иметь сложность  $O(\sqrt{M})$ .

Пример решения на языке Python.

```
n = int(input())
m = int(input())
if (1 + n) * n // 2 < m :
    print(0)
else :
    while m > n :
        print(n)
        m -= n
        n -= 1
    print(m)
```