

## Задача А. Отпуск

Если  $x + y < n$   $x + y < n$ , то было  $n - x - y$   $n - x - y$  дней без шторма и холодов, если же  $x + y \geq n$   $x + y \geq n$ , значит не было ни одного такого дня.

Пример кода на языке Python:

```
n = int(input())
x = int(input())
y = int(input())
if x + y < n:
    print(n - x - y)
else:
    print(0)
```

## Задача В. Простая игра

Минимальное число ходов будет, если все время будет выпадать 6. В этом случае число ходов будет равно  $n/6$   $n / 6$ , округленное вверх до ближайшего целого. В большинстве языков по умолчанию делается с округлением вниз, чтобы сделать округление вверх, проще всего использовать формулу  $(n + 5)/6$   $(n + 5) / 6$ , при этом деление делается с округлением вниз. Максимальное число ходов будет очевидно равно  $n$ .

Пример кода на языке Python:

```
n = int(input())
print((n + 5) // 6, n);
```

## Задача С. Сортировка сыра

Есть много способов решать эту задачу. Самый простой и эффективный способ такой. Пойдем справа налево. Последний столбик уменьшать смысла нет, так как он должен быть самым высоким.

Посмотрим на предпоследний столбик. Если он меньше, чем последний, то его опять же нет смысла уменьшать, а если выше — то его следует уменьшить до той же высоты. И так далее, каждый раз сравниваем столбик со следующим, если он выше, то уменьшаем его, чтобы они стали равны.

Пример кода на языке Python:

```
n = int(input())
a = [int(x) for x in input().split()]

res = 0
for i in range(n - 2, -1, -1):
    if a[i] > a[i + 1]:
        res += a[i] - a[i + 1]
        a[i] = a[i + 1]
```

```
print(res)
```

## Задача D. Верные утверждения

Эта задача вызвала, к сожалению, больше вопросов у участников, чем предполагало жюри олимпиады. Давайте разбираться, что в ней происходит. Есть  $n$  утверждений вида "Ровно  $a_i$  из этих утверждений верны". Нужно найти максимальное число верных утверждений, которое может быть среди них. Пусть среди них ровно  $k$  верных утверждений, тогда все утверждения, для которых  $a_i = k$  верны, а остальные нет. Такое может быть, если утверждений, для которых  $a_i = k$  ровно  $k$ . Посчитаем для каждого  $k$ , сколько всего утверждений, в которых  $a_i = k$ , после чего переберем подходящие  $k$  и выберем максимальное.

Отдельной проблемой для многих участников стал случай, когда ответ равен 0. Это возможно, если ни один ответ больше 0 не может быть правильным, и при этом нет утверждений, в которых  $a_i = 0$ . Если же у нас есть утверждения, в которых  $a_i = 0$ , то ответ 0 также не может быть правильным, поэтому надо вывести  $-1$ .

Пример кода на языке Python:

```
n = int(input())
a = [int(x) for x in input().split()]
c = [0] * (n + 1)
for x in a:
    c[x] += 1

res = -1
for k in range(n + 1):
    if c[k] == k:
        res = k

print(res)
```

## Задача E. Задачи для олимпиады

Довольно сложная задача. Можно решать ее, разбирая множество частных случаев, но так достаточно легко допустить ошибку. Чтобы упростить решение, можно зафиксировать один из параметров так, чтобы остальные вычислялись легко. Например, можно перебрать число туров, которые мы хотим провести. Если это число равно  $x$ , то как проверить, хватит ли у нас задач? Посмотрим на первую задачу. Это должна быть задача сложности 1 или 2. Понятно, что надо в первую очередь брать задачи сложности 1, потому что задачи сложности 2 можно использовать еще и на второй позиции. Тогда нам надо использовать  $q = \max(0, x - a[0])$  задач сложности 2. Теперь так же посмотрим, хватает ли нам задач на вторую позицию, при этом в первую очередь используем задачи сложности 2 и 3, а затем уже 4. В конце посмотрим, хватает ли нам задач сложности 4 и 5 на третью позицию.

Пример кода на языке Python:

```
a = [int(input()) for i in range(5)]
x = 0
while True:
```

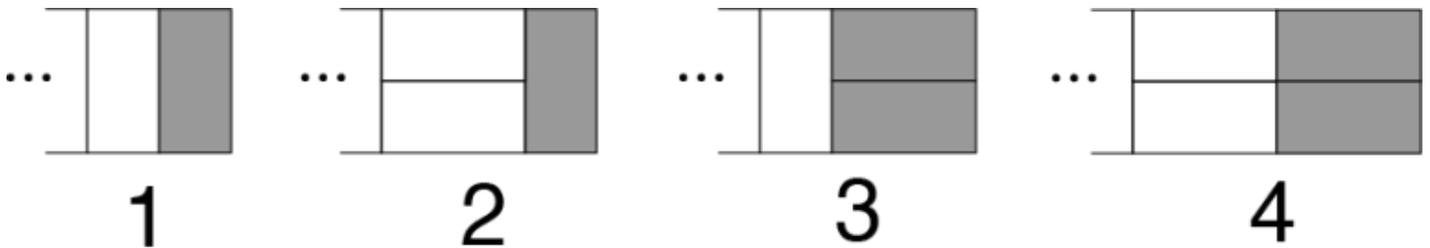
```

x += 1
if a[0] + a[1] < x: # хватает ли задач на позицию 1
    break
q = max(0, x - a[0])
if a[1] + a[2] + a[3] - q < x: # хватает ли задач на позицию 2
    break
w = max(0, x - a[1] - a[2] + q)
if a[3] + a[4] - w < x: # хватает ли задач на позицию 3
    break
print(x - 1)

```

## Задача F. Покраска доминошек

Пойдем слева направо и будем считать, сколько есть вариантов покрасить каждую доминошку. Каждый раз к картинке будет добавляться либо одна вертикальная доминошка, либо две горизонтальных. Рассмотрим четыре случая:



- Случай 1: добавили вертикальную доминошку, и перед ней стояла тоже вертикальная. Тогда есть два способа покрасить новую доминошку.
- Случай 2: добавили вертикальную доминошку, и перед ней стояли две горизонтальные. Тогда есть только один способ покрасить новую доминошку.
- Случай 3: добавили две горизонтальные доминошки, и перед ними стояла вертикальная. Тогда есть два способа покрасить новые доминошки.
- Случай 4: добавили две горизонтальные доминошки, и перед ними стояли еще две горизонтальные. Тогда есть три способа покрасить новые доминошки.

Отдельно надо рассмотреть самые левые доминошки.

Пример кода на языке Python:

```

n = int(input())
s1 = input()
s2 = input()

i = 0
last = 0
res = 1
while i < n:
    if s1[i] == s2[i]:
        if last == 0:
            res *= 3 # первая вертикальная доминошка
        elif last == 1:

```

```
        res *= 2 # случай 1
last = 1
i += 1
else:
    if last == 0:
        res *= 6 # первые горизонтальные доминошки
    elif last == 1:
        res *= 2 # случай 3
    else:
        res *= 3 # случай 4
last = 2
i += 2

print(res)
```