

7-8 класс

1. «Площадь рамки» (100 баллов)

Для нахождения площади рамки нужно из площади внешнего квадрата вычесть площадь внутреннего квадрата.

Сторону внешнего квадрата можно найти по формуле: $n-d*2*(11-k)$.

Сторону внутреннего квадрата можно найти по формуле: $n-d*2*(11-k-1)$.

Площадь рамки будет: $(n-d*2*(11-k))*(n-d*2*(11-k)) - (n-d*2*(11-k-1))*(n-d*2*(11-k-1))$.

В качестве ответа будет засчитана эта или любая другая эквивалентная формула.

2. «Удобное расположение исследовательской станции» (100 баллов)

Ответ к данной задаче содержится в файлах:

```
baza_cor01.txt
```

```
39
```

```
baza_cor02.txt
```

```
2043
```

Эта задача не требует написания программы, но прилагается и программная реализация решения данной задачи в файле **z2_baza.py**.

Также в файле **z2_baza.xlsx** предлагается решение данной задачи в MSExcel.

Числа из множества $\{0, 1, 5, 7\}$ обладают свойством, что их сумма равна 13, но никакие другие четыре числа, равные элементам этого множества, не дают такую же сумму. Таким образом, для решения задачи достаточно перебрать все квадраты 2×2 и посчитать количество квадратов, сумма в которых равна 13.

В Excel для подсчета суммы чисел в квадратах 2×2 можно использовать дополнительную таблицу. А функцией ЕСЛИ определить, подходит ли данный квадрат для строительства базы.

```
=ЕСЛИ(СУММ(baza_in01!A1:B2)=13;1;0)
```

Останется просуммировать количество 1 и получить ответ.

То же самое можно сделать с помощью языка программирования.

3. «Перемещение букв»(100 баллов)

Ответ к данной задаче содержится в файлах:

```
bukva_cor01.txt
```

```
E DOWN
R UP
D UP
A RIGHT
A RIGHT
A DOWN
S LEFT
S LEFT
S LEFT
S LEFT
A UP
A RIGHT
A RIGHT
D DOWN
D RIGHT
D RIGHT
R LEFT
R DOWN
E LEFT
```

```
bukva_cor02.txt
```

```
R UP
R UP
I LEFT
I UP
P RIGHT
P RIGHT
P DOWN
P DOWN
K LEFT
K LEFT
```

```

K LEFT
A LEFT
A LEFT
A DOWN
I DOWN
I RIGHT
I RIGHT
I RIGHT
R DOWN
R DOWN
R RIGHT
R RIGHT
A UP
A LEFT
P UP
P UP

```

Задача на умение записать алгоритм для исполнителя с заданной системой команд. Если алгоритм содержит не более чем 20 команд (для первой задачи) или 26 команд (для второй задачи) и правильно перемещает карточки с буквами согласно ограничениям в условии задачи, то участник получает полный балл за задачу. Если алгоритм содержит более указанного количества команд, но при этом алгоритм решает поставленную задачу, то участник получает 40 баллов. Если алгоритм правильно расставляет только некоторые буквы, то участник получает частичный балл, в зависимости от правильно поставленных букв.

4. «Дом с привидениями» (100 баллов)

Задача на проверку условий. Для каждого привидения нужно написать свое условие, которое определит, сможем ли мы его увидеть.

Чтобы увидеть Каспера, требуется, чтобы дата была четная и одно из двух условий: либо час нечетный, либо час четный и минуты не равны 00 (тогда, поскольку экскурсия длится час, в момент окончания экскурсии часы будут показывать нечетное значение и Каспера удастся увидеть).

$$d \% 2 == 0 \text{ and } (hh \% 2 != 0 \text{ or } mm != 0)$$

Чтобы увидеть Слимера требуется только, чтобы день был нечетный.

$$d \% 2 != 0$$

Поскольку для того, чтобы увидеть Битлджус, требуется, чтобы день, час или минуты оканчивались на 5, а во время часовой экскурсии часы несколько раз показывают время, когда минуты оканчиваются на 5, то его Вы сможете увидеть в любом случае.

Дядюшку Ау можно увидеть только, если день, год или час содержат цифру 7.

$$("7" \text{ instr}(d) \text{ or } "7" \text{ instr}(y) \text{ or } "7" \text{ in str}(hh) \text{ or } "6" \text{ in str}(hh) \text{ and } mm != 0)$$

Пивза (Peeves) можно увидеть, когда день кратен 3 и час кратен 3 или часы дают в остатке от деления на 3 число 2, а минуты не нулевые. В последнем случае во время часовой экскурсии будет момент, когда часы будут кратны 3.

$$d \% 3 == 0 \text{ and } (hh \% 3 == 0 \text{ or } mm != 0 \text{ and } hh \% 3 == 2)$$

Призрак прошлого Рождества появляется в рождественские дни с 24 декабря по 7 января.

$$m == 12 \text{ and } d >= 24 \text{ or } m == 1 \text{ and } d <= 7$$

Проверять условия и выводить имена привидений можно в любом порядке.

Так как Битлджус можно увидеть на любой экскурсии, то «No» выводить никогда не нужно.

```

d = int(input())
m = int(input())
y = int(input())
hh = int(input())
mm = int(input())
if d % 2 == 0 and (hh % 2 != 0 or mm != 0):
    print("Casper")
if d % 2 != 0:
    print("Slimer")
print("Beetlejuice")
if "7" in str(d) or "7" in str(y) or "7" in str(hh):
    print("Ow")
if d % 3 == 0 and (hh % 3 == 0 or mm != 0 and hh % 3 == 2):
    print("Peeves")
if m == 12 and d >= 24 or m == 1 and d <= 7:
    print("Christmas")

```

5. «Конкурсы» (100 баллов)

Задача на моделирование ситуации циклом. Нужно последовательно проводить суммирование, пока сумма не превысит указанного в условии числа B . Необходимо обратить внимание, что если последнее слагаемое, которое удалось добавить к сумме, равно $A + K$, то количество конкурсов, в которых принял участие Алексей, будет $K + 1$, поскольку в первом конкурсе он получил $A + 0$ наклеек.

```
a = int(input())
b = int(input())
s = a
if a > b:
    print(0)
else:
    k = 1
    while s + a + k <= b:
        s += a + k
        k += 1
    print(k)
```

6. «Прогулка по магазинам» (100 баллов)

Галя может сразу выйти к первому магазину, который она хочет посетить на улице, и посетить K магазинов подряд. При этом она пройдет расстояния между этими магазинами. Промежутков между K магазинами будет $K - 1$. Задача сводится к поиску суммы $K - 1$ подряд идущих элементов массива.

Поскольку расстояние, пройденное Галей по улице, должно быть минимальным, то она выйдет сразу к нужному магазину, а после посещения последнего уйдет с улицы. a_0 и a_{N+1} учитывать при нахождении суммы не нужно. По этим участкам Галя точно может не проходить.

Можно для каждого начального элемента каждый раз явно суммировать $K - 1$ элемент и среди полученных сумм искать минимальную. Такое решение может получить 80 баллов.

```
n = int(input())
a = []
for i in range(n + 1):
    a.append(int(input()))
k = int(input())
sminn = 1000 * 1000000
for i in range(1, n + 1 - k + 1):
    s = 0
    for j in range(i, i + k - 1):
        s += a[j]
    sminn = min(sminn, s)
print(sminn)
```

Сложность данного алгоритма $O(N^2)$.

Можно двумя вложенными циклами перебирать номера первого и последнего посещенных Галей магазинов. Для каждой пары нужно проверять условие, что они отличаются на $K - 1$. После этого третьим вложенным циклом суммировать элементы между этими индексами, включая элементы с выбранными индексами. Среди найденных сумм искать минимальную. Сложность данного алгоритма $O(N^3)$. Данное решение может набрать 60 баллов.

Полный балл получит решение, которое имеет сложность $O(N)$. Найдем сумму $a_1 + a_2 + \dots + a_{K-1}$ элементов. Чтобы из этой суммы получить сумму $a_2 + a_3 + \dots + a_K$ элементов достаточно из предыдущей суммы вычесть a_1 и добавить a_K . Аналогичным образом можно, скорректировав предыдущую сумму $K - 1$ элементов, получить сумму следующих $K - 1$ элементов. Среди всех найденных сумм найдем минимальную.

```
n = int(input())
a = []
for i in range(n + 1):
    a.append(int(input()))
k = int(input())

s = sum(a[1: k])
sminn = s
for i in range(1, n - k + 1):
    s -= a[i]
    s += a[i + k - 1]
    sminn = min(sminn, s)
print(sminn)
```