

# Использование RUTHON для решения заданий КЕГЭ

Задание 27

## Задание 27.

Обработка данных, вводимых из файла в виде последовательности чисел.

- **Знать:**

- как прочитать данные из файла
- основы комбинаторики
- динамическое программирование

# Типы задач

- Нахождение максимальной последовательности, сумма которой
  - не кратна заданному числу
  - кратна заданному числу
- Нахождение количества последовательностей (множеств)

- Нахождение максимальной последовательности, сумма которой **не кратна заданному числу**
  - ❑ Найти максимально возможную сумму последовательности
  - ❑ Найти минимальную разницу (разность) между двумя элементами пары

*Имеется набор данных, состоящий из пар положительных целых чисел.*

*Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно. Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.*

### ***Входные данные***

*Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество пар  $N$  ( $1 \leq N \leq 100000$ ). Каждая из следующих  $N$  строк содержит два натуральных числа, не превышающих 10 000.*

*Пример организации исходных данных во входном файле:*

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

*Для указанных входных данных значением искомой суммы должно быть число 32.*

*В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.*

```
F = open( "27.txt" )
N = int( F.readline() )
s, dMin = 0, 10001
for i in range(N):
    a, b = map( int, F.readline().split() )
    s += max( a, b )
    d = abs( a-b )
    if d % 3 > 0:
        dMin = min( d, dMin )
if s % 3 != 0:
    print( s )
else:
    print( s-dMin )
```

- Нахождение максимальной последовательности, сумма которой **кратна заданному числу**
  - Найти максимально возможную сумму последовательности
  - Найти значение, которое имеет тот же остаток от деления, что и максимальная сумма

**Значение, которое имеет тот же остаток от деления на заданное число, что и максимальная сумма**

- Разность между двумя значениями в одной из пар**
- Накапливаемое значение из нескольких пар**



*Имеется набор данных, состоящий из пар положительных целых чисел.*

*Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел **делилась на 6** и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно. Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.*

### ***Входные данные***

*Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество пар  $N$  ( $1 \leq N \leq 100000$ ). Каждая из следующих  $N$  строк содержит два натуральных числа, не превышающих 10 000.*

*Пример организации исходных данных во входном файле:*

```
3
7 4
11 9
5 23
```

*Для указанных входных данных значением искомой суммы должно быть число 36 (выбраны числа 4, 9 и 23, их сумма 36 делится на 6). В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.*

```
F = open("27.txt")
D = 6
N = int(F.readline())
s = 0
dMin = [10001]*D
for i in range(N):
    a, b = map(int, F.readline().split())
    s += max(a, b)
    d = abs(a-b)
    r = d % D
    if r > 0:
        dMinNew = dMin
        for k in range(1, D):
            r0 = (r + k) % D
            dMinNew[r0] = min(d+dMin[k], dMinNew[r0])
        dMinNew[r] = min(d, dMinNew[r])
        dMin = dMinNew
F.close()
if s % D == 0:
    print(s)
else:
    print(s - dMin[s % D])
```

*В файле записана последовательность натуральных чисел. Гарантируется, что все числа различны. Рассматриваются всевозможные непустые подмножества, состоящие из элементов последовательности. Необходимо найти количество подмножеств, в которых сумма элементов кратна 12.*

### ***Входные данные***

*Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество пар  $N$  ( $1 \leq N \leq 100000$ ). Каждая из следующих  $N$  строк содержит одно натуральное число, не превышающих  $10^6$ .*

*Пример организации исходных данных во входном файле:*

**4**

**5**

**7**

**12**

**23**

*Для указанных данных можно выбрать следующие подмножества: {12}; {5, 7}; {5, 7, 12}. Программа должна вывести количество этих множеств – 3. В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.*

```
f = open('271.txt')
n = int(f.readline())
k = [0]*12
for i in range(n):
    x = int(f.readline())
    k1 = k.copy()
    for i in range(12):
        k1[ (i+x) % 12 ] += k[i]
    k1[ x%12 ]+=1
    k = k1.copy()
f.close()
print(k[0])
```

Дана последовательность из  $N$  натуральных чисел. Рассматриваются все её непрерывные подпоследовательности, такие что сумма элементов каждой из них кратна  $k = 43$ . Найдите среди них подпоследовательность с максимальной суммой, определите её длину. Если таких подпоследовательностей найдено несколько, в ответе укажите количество элементов самой короткой из них.

### **Входные данные**

Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество пар  $N$  ( $1 \leq N \leq 10^8$ ). Каждая из следующих  $N$  строк содержит одно натуральное число, не превышающих 10000.

Пример организации исходных данных во входном файле:

7  
21  
13  
9  
19  
17  
26  
95

В этом наборе можно выбрать последовательности  $21+13+9$  (сумма 43) и  $17+26$  (сумма 43). Самая короткая из них,  $17 + 26$ , имеет длину 2. Для указанных программа должна вывести число 2. В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.

```
F = open("272.txt")
N = int( F.readline())
K = 43
tailSum = [0] + [None]*(K-1)
tailLen = [0]*K
maxSum, minLen = 0, 0
totalSum = 0
for i in range(1,N+1):
    x = int( F.readline() )
    totalSum += x
    r = totalSum % K
    if tailSum[r] != None:
        curSum = totalSum - tailSum[r]
        curLen = i - tailLen[r]
        if curSum > maxSum or (curSum == maxSum and curLen < minLen):
            maxSum = curSum
            minLen = curLen
    else:
        tailSum[r] = totalSum
        tailLen[r] = i
F.close()
print(minLen)
```

27

Дана последовательность целых чисел. Необходимо найти максимально возможную сумму её непрерывной подпоследовательности, в которой количество положительных чётных элементов кратно  $k = 30$ .

**Входные данные**

Первая строка входного файла содержит целое число  $N$  – общее количество чисел в наборе. Каждая из следующих  $N$  строк содержит одно число. Гарантируется, что общая сумма любой выборки заданных чисел не превышает  $2 \cdot 10^9$  по абсолютной величине.

Вам даны два входных файла (A и B), каждый из которых имеет описанную выше структуру. В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.

Ответ:

--	--

```

k=30
f=open("27-A.txt")
#f=open("27-B.txt")
n=int(f.readline())
count=0
maxS=-2*10**9
Mostat=[2*10**9]*k
s=0
for i in range(n):
    a=int(f.readline())
    s+=a
    if a>0 and a%2==0:
        count+=1
    ostat=count%k
    if ostat==0 and s>maxS:
        maxS=s
    if Mostat[ostat]!=2*10**9 and s-Mostat[ostat]>maxS:
        maxS=s-Mostat[ostat]
    Mostat[ostat]=min(Mostat[ostat],s)
f.close()
print(maxS)

```