

# Особенности решения задач 25 и 26 компьютерного ЕГЭ по информатике

**К.Ю. Поляков**

вебинар для учителей информатики г. Сочи  
24 марта 2021 года

## 25. Пример

---

**(Демо-2021)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку  $[174457; 174505]$ , числа, имеющие ровно **два различных натуральных делителя**, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

## 25. Общий подход

---

- 1) Пишем решение «в лоб».
- 2) Если получили ответ, то СТОП.
- 3) Оптимизируем.
- 4) Переходим к шагу 2.



Не нужно оптимизировать  
без необходимости!

## 25. Решение

```
var startN:= 174457000;  
var endN:= 174505000;
```

```
##  
var startN:= 174457;  
var endN:= 174505;  
for var x:=startN to endN do begin  
    var count:= 0;  
    var divs:= |0, 0|;  
    for var d:=2 to x-1 do  
        if x.Divs(d) then begin  
            count += 1;  
            if count > 2 then break;  
            divs[count-1] := d;  
        end;  
    if count = 2 then  
        Println( divs[0], divs[1] );  
end;
```



Как ускорить?

## 25. Ускорение

```
for var d:=2 to x div 2 do begin
```

```
    ...
end;
```



Сложность алгоритма не меняется!  
 $O(N^2)$

Делители в парах:

$$x = d \cdot g \quad (d \leq g) \Rightarrow d \leq \sqrt{x}$$



Проблема: вещественное  $\sqrt{x}$  !

$$g = \frac{x}{d}$$

$$d \leq \sqrt{x} \Rightarrow d^2 \leq x$$



Проблема: полные квадраты!

$$d = g \Rightarrow \text{один делитель}$$

## 25. Квадратный корень

```
##  
var x := 1000000000000;  
while x < 10000000000000 do begin  
  var sqrtX := sqrt(x);  
  if x <> sqrtX*sqrtX then  
    Println(x, x-sqrtX*sqrtX);  
  x := x + 1;  
end;
```



Ошибка  $\pm 1,53 \times 10^{-5}$ !



trunc? round? ceil?

## 25. Квадратный корень

```
for var d:=2 to round(sqrt(x)) do begin  
    ...  
end;
```

1) полный квадрат

$$\left. \begin{array}{l} \sqrt{25} \approx 5 - \varepsilon \\ \sqrt{25} \approx 5 + \varepsilon \end{array} \right\} \xrightarrow{\text{round}} 5$$

2) два множителя с разностью 1

$$\left. \begin{array}{l} \sqrt{30} \approx 5,477 - \varepsilon \\ \sqrt{30} \approx 5,477 + \varepsilon \end{array} \right\} \xrightarrow{\text{round}} 5$$



**round** здесь округлит к меньшему!

## 25. Квадратный корень

---

Без вещественных чисел:

$$d \leq \sqrt{x} \Rightarrow d^2 \leq x$$

```
var d:= 2;  
while d*d <= x do begin  
    ...  
    d += 1;  
end;
```



## 25. Список делителей

```
##
var startN := 174457;
var endN := 174505;
for var x:=startN to endN do begin
  var divs:= new List<integer>;
  for var d:=2 to round(sqrt(x)) do
    if x.Divs(d) then begin
      divs.Add(d);
      if d <> x div d then
        divs.Add(x div d);
      if divs.Count > 2 then break;
    end;
  if divs.Count = 2 then
    Println( divs[0], divs[1] );
end;
```

$d*d \neq x$

## 25. Простые числа

```
function IsPrime (x: integer) : boolean;
begin
  Result:= False;
  if x <= 1 then Exit;
  var d:= 2;
  while d*d <= x do begin
    if x.Divs (d) then Exit;
    d += 1;
  end;
  Result:= True;
end;
```

## 25. Список простых чисел

---

```
var primes := new List<integer>;  
for var i:=1 to 1000000 do  
    if IsPrime(i) then  
        primes.Add(i);  
Print( primes.Count );
```



Время 0,3 с!

## 25. Пример

---

**(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [194441; 196500] **простые числа, оканчивающиеся на 93.**

## 25. Решение «в лоб»

---

```
var startN := 194441;  
var endN := 196500;  
for var x:=startN to endN do  
    if (x mod 100 = 93) and  
        IsPrime(x) then  
        Println(x);
```

## 25. Оптимизация

```
var startN:= 194493 ;  
var endN:= 196500 ;  
var x:= startN;  
while x <= endN do begin  
    if IsPrime(x) then Println(x) ;  
    x += 100 ;  
end;
```

первое, которое  
оканчивается на 93

## 25. Пример

---

Рассматриваются целые числа, принадлежащих числовому отрезку  $[631632; 684934]$ , которые представляют собой **произведение двух различных простых делителей**. Найдите такое из этих чисел, у которого два простых делителя больше всего отличаются друг от друга.

## 25. Решение «в лоб»

```
var startN := 631632;  
var endN := 684934;  
var maxDiff := 0;  
var xMaxDiff := 0;  
for var x:=startN to endN do  
  for var d:=2 to round(sqrt(x))-1 do  
    if x.Divs(d) and IsPrime(d) and  
      IsPrime(x div d) and  
      (x div d - d > maxDiff) then  
      begin  
        maxDiff := x div d - d;  
        xMaxDiff := x;  
      end;  
Println( xMaxDiff, maxDiff );
```

```
var startN:= 63163200;  
var endN:= 68493400;
```



Как ускорить?



## 25. Оптимизация

```
for var x:=startN to endN do
  for var d:=2 to round(sqrt(x))-1 do
    if x.Divs(d) then begin
      if IsPrime(x div d) and
        (x div d - d > maxDiff) then begin
        maxDiff := x div d - d;
        xMaxDiff := x;
      end;
      break;
    end;
  end;
```

~~IsPrime(d)~~  
первый d всегда  
простой!



Пара «наименьший-наибольший» имеет наибольшую разность!

## 25. Используем только простые

---

Список возможных меньших простых делителей:

```
var primes := new List<integer>;  
for var i:=1 to round(sqrt(endN)) do  
    if IsPrime(i) then  
        primes.Add(i);
```

## 25. Используем только простые

```
for var x:=startN to endN do
  foreach var d in primes do
    if x.Divs(d) then begin
      if IsPrime(x div d) and
        (x div d - d > maxDiff) then
        begin
          maxDiff := x div d - d;
          xMaxDiff := x;
        end;
      break;
    end;
  end;
```

```
var startN:= 63163200;
var endN:= 68493400;
```



16,7 с!

## 17. Пример

---

Назовём натуральное число подходящим, если ровно два из его делителей входят в список (7, 11, 13, 19). Найдите все подходящие числа, принадлежащих отрезку [20 000; 30 000]

В ответе запишите два целых числа: сначала количество, затем среднее арифметическое всех найденных чисел (только целую часть).

### Проблемы:

- 1) ровно два из его делителей входят в список
- 2) среднее арифметическое всех найденных чисел (сумма может быть очень велика!)

## 17. Ровно два делителя

```
var startN := 20000;  
var endN   := 30000;  
var count  := 0;  
for var x:=startN to endN do begin  
    var divs := | integer(x mod 7 = 0),  
                 ord(x mod 11 = 0),  
                 ord(x.Divs(13)),  
                 1-sign(x mod 19) |;  
    if divs.Sum = 2 then  
        count += 1;  
end;  
Println( count );
```

МОЖНО  
по-разному!



Как быть с суммой?



- 1) `int64`
- 2) `double`

## 25. Пример

---

(Статград) Найдите все натуральные числа, принадлежащие отрезку  $[289123456; 389123456]$  и имеющие ровно три нетривиальных делителя. Для каждого найденного числа запишите в ответе его наибольший нетривиальный делитель.

**Проблемы:**

долго считает...

## 25. Решение «в лоб»

```
var startN := 289123456;  
var endN   := 389123456;  
for var x:=startN to endN do begin  
    var divs := new List<integer>;  
    for var d:=2 to x-1 do  
        if x.Divs(d) then divs.Add(d);  
    if divs.Count = 3 then  
        Println( x );  
end;
```



Как ускорить?

## 25. Только квадраты

Три (нечётное число) нетривиальных делителя – полный квадрат!

```
var startN := 289123456;  
var endN   := 389123456;  
for var sqrtX:=trunc(sqrt(startN)) to  
           ceil(sqrt(endN)) do begin  
    var x := sqrtX*sqrtX;  
    var divs := new List<integer>;  
    for var d:=2 to x-1 do  
        if x.Divs(d) then divs.Add(d);  
    if divs.Count = 3 then  
        Println(x);  
end;
```



Как ускорить?



## 25. Основная теорема арифметики

---

Любое число единственным способом представляется в виде произведения простых чисел:

$$n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$$

Число нетривиальных делителей:

$$\delta = (k_1 + 1)(k_2 + 1) \dots (k_m + 1) - 2$$

Если  $\delta = 3$ :

$$(k_1 + 1)(k_2 + 1) \dots (k_m + 1) = 5$$

$$k_1 = 4, \quad k_2 = k_3 = \dots = k_m = 0$$

## 25. Только четвёртые степени

```
var startN := 289123456;  
var endN   := 389123456;  
for var qX:=trunc(sqrt(sqrt(startN))) to  
        ceil(sqrt(sqrt(endN))) do begin  
    if IsPrime(qX) then  
        Println( qX*qX*qX*qX );  
end;
```

 0,016 с!

## 25. Готовые функции


---

**(Демо-2021)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку  $[174457; 174505]$ , числа, имеющие ровно **два различных натуральных делителя**, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

## 25. Модуль school

---

```
##  
uses school;  
for var x:=174457 to 174505 do begin  
    var divs := x.Divisors;  
    if divs.Count = 4 then  
        Println( divs[1], divs[2] );  
end;
```



## 25. Функциональный стиль

---

```
##  
uses school;  
(174457..174505)  
  .Select( x->x.Divisors )  
  .Where( x->x.Count = 4 )  
  .Select( x->(x[1],x[2]) )  
  .PrintLines;
```

(174457..174505) последовательность чисел

Презентации С.С. Михалковича:

<http://www.pascalabc.net/downloads/Presentations/Tutorials/Sequences.pdf>

<http://www.pascalabc.net/downloads/Presentations/Tutorials/ProcFuncLambdas.pdf>

## 25. Последовательность

```
function mySeq( a, b: integer ):
    sequence of integer ;
begin
    while a <= b do begin
        yield a;
        a += 1;
    end;
end;
begin
    foreach var x in mySeq(10,20) do
        Print(x) ;
end.
```

10 11 12 13 14 15 16 17 18 19 20

## 25. Функциональный стиль

```
(10..20).Select( x->x.Divisors ).PrintLines;
```

заменить каждый элемент последовательности  
на список его делителей

[1, 2, 5, 10]

все делители 10

[1, 11]

11

[1, 2, 3, 4, 6, 12]

12

[1, 13]

13

[1, 2, 7, 14]

[1, 3, 5, 15]

[1, 2, 4, 8, 16]

[1, 17]

..

## 25. Функциональный стиль

---

```
(10..20).Select( x->x.Divisors )  
    .Where( x->x.Count = 4 ).PrintLines;
```

отобразить те элементы списка, где количество делителей равно 4

[1, 2, 5, 10]

10

[1, 2, 7, 14]

14

[1, 3, 5, 15]

15



## 25. Функциональный стиль

```
(10..20).Select( x->x.Divisors )  
    .Where( x->x.Count = 4 )  
    .Select( x-> (x[1],x[2]) ).PrintLines;
```

заменить каждый элемент списка на пару  
(**кортеж**), состоящую из двух нетривиальных  
делителей

(2, 5)

10

(2, 7)

14

(3, 5)

15

## 25. Пример

---

**(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку  $[194441; 196500]$  простые числа, **оканчивающиеся на 93**.

## 25. Функциональный стиль

```
##  
uses school;  
(194441..196500)  
  .Where ( x-> (x mod 100 = 93) and  
           x.IsPrime )  
  .Println;
```

```
##  
uses school;  
( 194493 ..196500) .Step(100)  
  .Where ( x->x.IsPrime )  
  .Println;
```

## 17. Пример

---

Назовём натуральное число подходящим, если ровно два из его делителей входят в список (7, 11, 13, 19). Найдите все подходящие числа, принадлежащих отрезку [20 000; 30 000]

В ответе запишите два целых числа: сначала количество, затем среднее арифметическое всех найденных чисел (только целую часть).

## 25. Функциональный стиль

```
##
uses school;
var selected :=
  (20000..30000)
  .Select( x->x.Divisors )
  .Select( x->(x.Last,
              integer(7 in x)+
              ord(...)) integer(11 in x)+
              integer(13 in x)+
              integer(19 in x)) )
  .Where( x-> x[1] = 2 )
  .Select( x->x[0] );
Println( selected.Count,
         trunc(selected.Average) );
```

## 25. Функциональный стиль

```
(70..91).Select( x->x.Divisors ).PrintLines;
```

заменить каждый элемент последовательности  
на список его делителей

```
[1, 2, 5, 7, 10, 14, 35, 70]
```

все делители 70

```
[1, 71]
```

```
[1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72]
```

```
[1, 73]
```

```
[1, 2, 37, 74]
```

```
[1, 3, 5, 15, 25, 75]
```

```
[1, 2, 4, 19, 38, 76]
```

```
..
```

## 25. Функциональный стиль

```
(70..91).Select( x->x.Divisors )
    .Select( x->(x.Last,
        integer(7 in x)+
        integer(11 in x)+
        integer(13 in x)+
        integer(19 in x) ) )
    .Println;
```

построить кортежи:

( число,  
количество делителей из [7,11,13,19])

(70,1) (71,0) (72,0) (73,0)

(74,0) (75,0) (76,1) (77,2)

..

## 25. Функциональный стиль

---

```
(70..91).Select( x->x.Divisors )  
    .Select( x->(x.Last, ...) )  
    .Where( x-> x[1] = 2 )  
    .Println;
```

отобрать те, где количество делителей из списка (**x[1]**) равно 2:

**(77, 2) (91, 2)**



## 25. Функциональный стиль

```
(70..91).Select( x->x.Divisors )
      .Select( x->(x.Last, ...) )
      .Where( x-> x[1] = 2 )
      .Select( x->x[0] )
      .Println;
```

ОСТАВИТЬ ТОЛЬКО САМИ ЧИСЛА (**x[0]**)

**77 91**

ВЫВЕСТИ КОЛИЧЕСТВО И СРЕДНЕЕ:

```
Println( selected.Count,
         selected.Average );
```

**2 84**

## 25. Функциональный стиль

```
##  
var z := (20000..30000)  
    .Select( x->(x,  
        |7,11,13,19|.Count(d->x.Divs(d)) ) )  
    .Where( x->x[1] = 2 )  
    .Select( x->x[0] );  
Println( z.Count, z.Average );
```

пары (число, кол-во делителей)

```
##  
var z := (20000..30000)  
    .Where( x->(|7,11,13,19|  
        .Count(d->x.Divs(d)) = 2 ) );  
Println( z.Count, z.Average );
```

два прохода по последовательности

## 25. Пример

---

(Статград) Найдите все натуральные числа, принадлежащие отрезку  $[289123456; 389123456]$  и имеющие ровно три нетривиальных делителя. Для каждого найденного числа запишите в ответе его наибольший нетривиальный делитель.

## 25. Функциональный стиль

```
##
uses school;
(trunc(sqrt(sqrt(289123456))) ..
 ceil(sqrt(sqrt(389123456))))
  .Where( x->x.IsPrime )
  .Select( x->x*x*x*x )
  .Println;
```

## 26. Сортировка

---

**(Демо-2021)** Раз в неделю создаёт архив файлов. Объём диска, может быть меньше, чем суммарный объём файлов. По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

## 26. Решение в Excel

---

<https://kpolyakov.spb.ru/download/ege26.doc>

А. Сидоров:

[www.youtube.com/watch?v=LwTZAHsno0k](http://www.youtube.com/watch?v=LwTZAHsno0k)

## 26. Решение

```
##
Assign(input, '26.txt');
var (V, n) := ReadInteger2;
var a := ReadArrInteger(n); a.Sort;

var i:=0;
while (i <= a.High) and (V >= a[i]) do begin
    V -= a[i];
    i += 1;
end;
i.Println;

V += a[i-1];
while (i <= a.High) and (V >= a[i]) do
    i += 1;
a[i-1].Print;
```

## 26. Сортировка

```
##  
var A := | 27, 19, 21, 33 |;  
A.Sort;  
A.Println;
```

$< 0 \Rightarrow x < y$   
 $= 0 \Rightarrow x = y$   
 $> 0 \Rightarrow x > y$

19 21 27 33

```
A.Sort( (x,y) -> x mod 10 - y mod 10 );  
A.Println;
```

21 33 27 19

```
A.Sort( (x,y) -> y mod 10 - x mod 10 );  
A.Println;
```

19 27 33 21



**.Sort** переставляет элементы!



## 26. Сортировка

```
##  
uses school;  
var A := | 41, 19, 21, 33 |;  
A.Sort( (x,y) -> x.Digits.Sum - y.Digits.Sum );  
A.Println;  
A.Sort( (x,y) -> y.Digits.Sum - x.Digits.Sum );  
A.Println;
```

по сумме цифр

```
21 41 33 19  
19 33 41 21
```

## 26. Сортировка (последовательности)

```
##  
var A := | 27, 19, 25, 33 |;  
A.Order.Println;
```

19 25 27 33

```
A.OrderDescending.Println;
```

33 27 25 19



`.Order` НЕ переставляет элементы и строит последовательность!

```
A.Sort; ↔ A := A.Order.ToArray;
```

## 26. Сортировка (последовательности)

```
##
```

```
var A := | 27, 19, 25, 33 |;  
A.OrderBy( x->x mod 10 ).Println;
```

по последней  
цифре

```
33 25 27 19
```

```
A.OrderByDescending( x->x mod 10 ).Println;
```

```
19 27 25 33
```

## 26. Сортировка (последовательности)

```
##  
uses school;  
var A := | 41, 19, 21, 33 |;  
A.OrderBy( x->x.Digits.Sum ).Println;  
A.OrderByDescending(  
    x->x.Digits.Sum ).Println;
```

по сумме  
цифр

```
21 41 33 19  
19 33 41 21
```

## 26. Пример

---

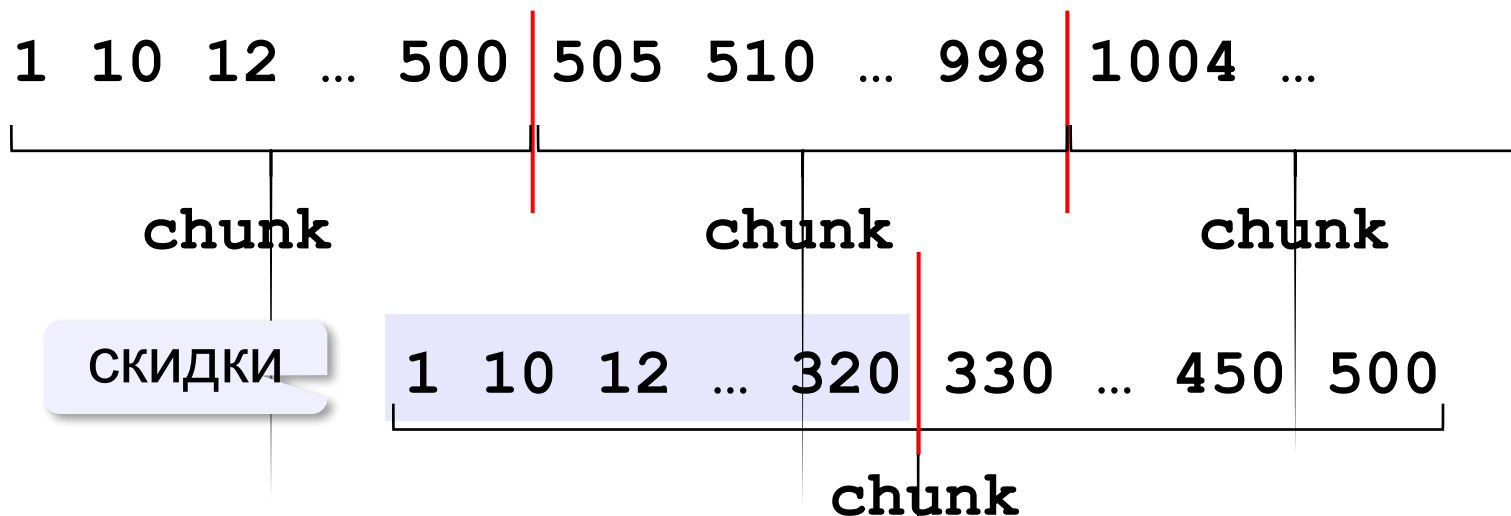
**(Е. Джобс)** В магазине проводят акция – каждый второй товар со скидкой 50%. При этом в акции участвуют только те товары, цены которых попадают в одну ценовую категорию. Каждая ценовая категория включает 500 целых значений: 1-500, 501-1000, 1001-1501 и т.д. Например, при наличии в чеке только позиций с ценами 300 и 1000 предложение акции не работает.

Необходимо распределить товары в чеке таким образом, чтобы итоговая цена всех товаров была максимально выгодной для магазина. В качестве ответа вывести полученную сумму скидки для всего чека и конечную стоимость самого дорогого проданного по акции товара. В случае получения нецелых значений привести только целые части найденных чисел.

## 26. Решение на Python

```
with open("26-44.txt") as F:
    N = int(F.readline())
    data = []
    for i in range(N):
        data.append( int(F.readline()) )

data.sort()
# ... продолжение следует
```



## 26. Решение на Python

```
# ... продолжение
last = 500
discount, costMax = 0, 0

while data:
    chunk = [x for x in data if x <= last]
    if chunk:
        mid = len(chunk)//2
        if mid > 0:
            discount += 0.5*sum(chunk[:mid])
            costMax = 0.5*chunk[mid-1]
        data = data[len(chunk):]
    last+= 500

print( int(discount), int(costMax) )
```

## 26. Решение на PascalABC.NET

```
##  
Assign(input, '26-44.txt');  
var N := ReadInteger;  
var data := ReadArrInteger(N);  
  
data.Sort;  
// ... продолжение следует
```



## 26. Решение на PascalABC.NET

```
# ... продолжение
var (last, discount, costMax) := (500, 0.0, 0.0);
while data.Count > 0 do begin
    var chunk := data.Where(x->x<=last).ToArray;
    if chunk.Count > 0 then begin
        var mid := chunk.Count div 2;
        if mid > 0 then begin
            discount += 0.5*chunk[:mid].Sum;
            costMax := 0.5*chunk[mid-1];
        end;
        data := data.Where(x->x>last).ToArray;
    end;
    last += 500;
end;
Println( trunc(discount), trunc(costMax) );
```

## 26. Пример

---

(**А. Кабанов**) На складе лежат пакеты с углём различного веса и стоимости. **Вес** и **стоимость** записаны на каждом пакете как натуральные числа. Для транспортировки отбираются  **$K$**  пакетов с самой низкой ценой угля за единицу веса; при равной стоимости за единицу веса выбираются пакеты с большим весом. По заданной информации о пакетах с углём и количестве транспортируемых пакетов определите

- 1) суммарный вес угля в отправленных пакетах и
- 2) стоимость самого тяжёлого отправленного пакета.

## 26. Решение на Python

```
with open("26-k6.txt") as F:  
    data = F.readlines()
```

```
N, K = map(int, data[0].split())
```

```
del data[0]
```

```
data = data[:N]
```

строим массив пар  
(**ВЕС, СТОИМОСТЬ**)

```
pairs = []
```

```
for i in range(N):
```

```
    p = tuple( map(int, data[i].split()) )
```

```
    pairs.append( p )
```

```
# ... продолжение следует
```

## 26. Решение на Python

цена за  
единицу веса

```
# ... продолжение
pairs.sort( key =
    lambda x: (x[1]/x[0], -x[0]) )

selected = pairs[:K]

print( sum( x[0] for x in selected ) )

weight = [x[0] for x in selected]
ind = weight.index( max(weight) )
print( selected[ind][1] )
```

по убыванию  
веса

суммарный вес

СТОИМОСТЬ пакета с  
наибольшим весом

## 26. Решение на PascalABC.NET

```
##
Assign(input, '26-k6.txt');
var (N, K) := ReadInteger2;

var pairs := (1..N)
    .Select( i->ReadString.ToIntegers )
    .Skip(1).ToArray;

pairs.Sort( (x, y) ->
    x[1]/x[0]-y[1]/y[0] <> 0 ?
    sign(x[1]/x[0]-y[1]/y[0]) : y[0]-x[0] );

// ... продолжение следует
```

условие

если  
невернопо убыванию  
весаесли  
верноцена за  
единицу веса

## 26. Решение на PascalABC.NET

```
# ... продолжение
var selected := pairs[:K];
var weight :=
    selected.Select( x->x[0] ).ToArray;
weight.Sum.Println;
var ind := weight.IndexOf( weight.Max );
selected[ind][1].Println;
```

суммарный вес

СТОИМОСТЬ пакета с  
наибольшим весом

## 26. Пример

---

В текстовом файле записан набор натуральных чисел. Гарантируется, что все числа различны. Необходимо определить, сколько в наборе таких пар нечётных чисел, что их среднее арифметическое тоже присутствует в файле, и чему равно наибольшее из средних арифметических таких пар.

## 26. Решение на Python

```
with open("26.txt") as F:
    N = int(F.readline())
    data = [int(s) for s in F]

data.sort()

averages = []
for i in range(N-1):
    for j in range(i+1, N):
        if data[i] % 2 == 1 and data[j] % 2 == 1:
            s = data[i] + data[j]
            averages.append( s//2 )

averages.sort()

# ... продолжение следует
```

НАХОДИМ ВСЕ  
ПОДХОДЯЩИЕ СРЕДНИЕ



## 26. Решение на Python (**плохое**)

```
# ... продолжение
```

```
count = 0
```

```
ma = 0
```

```
for av in averages:
```

```
    if av in sa:
```

```
        count = count + 1
```

```
        ma = max( ma, av )
```

```
print( count, ma )
```

СЛОЖНОСТЬ  
 $O(N^2)$

## 26. Идея хорошего решения

9, 10, 14, 13, 8, 11

Все пары нечётных чисел:

(9, 13) (9, 11) (13, 11)

11 10 12

**!** Отсортируем!

8, 9, 10, 11, 13, 14

10 11 12

**!** Больше значение находится ближе к концу массива!

## 26. Решение на Python

```
# ... продолжение
selected = []
i = 0
for av in averages:
    while i < N and data[i] < av:
        i += 1
    if i < N and data[i] == av:
        selected.append(av)

print( len(selected) , selected[-1] )
```

## 26. Решение на PascalABC.NET

```
##
Assign(input, '26.txt');
var N:= ReadInteger;
var data:= ReadArrInteger(N);
data.Sort;

var averages := new List<integer>;
for var i:=0 to data.High-1 do
    for var j:=i+1 to data.High do
        if data[i].IsOdd and
            data[j].IsOdd then
            averages.Add((data[i]+data[j]) div 2);

averages.Sort;

// ... продолжение следует
```

## 26. Решение на PascalABC.NET

```
##
Assign(input, '26.txt');
var N:= ReadInteger;
var data:= ReadArrInteger(N);
data.Sort;

var averages := new List<integer>;
foreach var (x,y) in data.Combinations(2) do
    if x.IsOdd and y.IsOdd then
        averages.Add((x+y) div 2);

averages.Sort;

// ... продолжение следует
```

или так:

## 26. Решение на PascalABC.NET

```
# ... продолжение
var selected := new List<integer>;
var i := 0;
foreach var av in averages do begin
    while (i < N) and (data[i] < av) do
        i += 1;
    if (i < N) and (data[i] = av) then
        selected.Add(av);
end;

Print( selected.Count, selected.Last )
```

# Благодарности

---

Автор благодарит

**Алексея Богданова** (Alex Danov)

<https://www.youtube.com/c/AlexDanov>

**Станислава Михалковича**

<https://pascalabc.net>

за полезные замечания и предложения.

# Конец фильма

---

**ПОЛЯКОВ Константин Юрьевич**

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

[kpolyakov@mail.ru](mailto:kpolyakov@mail.ru)