

Вопросы методики
подготовки к
выполнению задания
26 К-ЕГЭ



Семашко Надежда Сергеевна,
учитель информатики и ИКТ
Лицей №6 им. академика Г.Н.Флерова, г. Дубна
ns.semashko@gmail.com

Характеристика 26 задания

- Согласно спецификации КИМ ЕГЭ 2021 задание 26 относится к **высокому** уровню сложности
- Проверяемые элементы – умение обрабатывать целочисленную информацию с использованием сортировки
- Оценивание:
 - ✓ 2 балла за верный ответ
 - ✓ 1 балл, если значения в ответе перепутаны местами или присутствует только одно правильное значение
 - ✓ 0 баллов в остальных случаях
- Рекомендуемое время выполнения 35 минут

Что нужно знать и уметь:

- *Работа с текстовыми файлами*

- a) исходные данные записаны в текстовом файле, их нужно прочитать в массив/список
- b) нужно знать методы для подключения и обработки текстовых файлов в используемом языке программирования

- *Обработка данных*

- a) знать особенности тех типов данных, которые участник экзамена выбирает для хранения данных (множества, кортежи, списки, строки, словари и как они сортируются)
- b) уметь сортировать данные с помощью встроенной сортировки по возрастанию/убыванию и **не рекомендуется** писать собственную реализацию алгоритмов сортировки
- c) уметь применять ключи сортировки, которые могут быть реализованы через функции (лямбда-функции), такие как сортировка по последней цифре числа, по длине строки и т.д.)

Что нужно знать и уметь:

- *Отбирать элементы массива/списка по условию*
- *Решение в ЭТ - > навыки работы с электронными таблицами*
 - a) сортировка и фильтрация данных
 - b) вычисления по формулам
 - c) отбор по условию

Задание 26 (демо 2022)

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов.

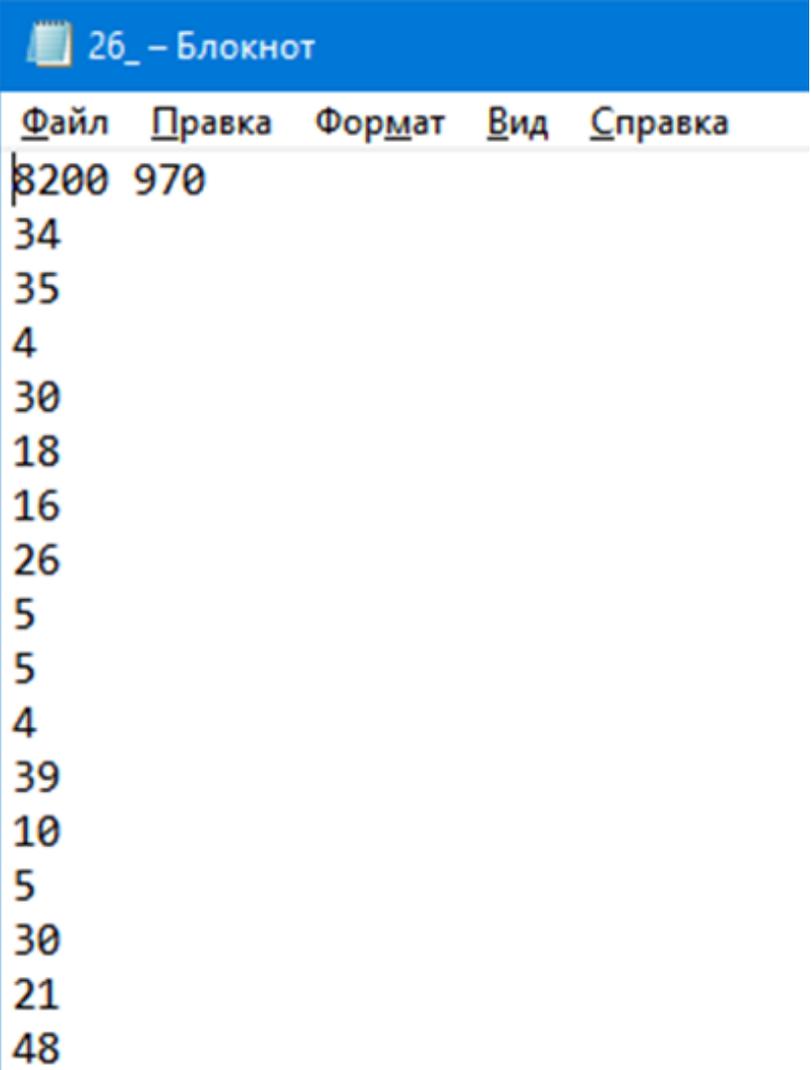
Известно, какой объём занимает файл каждого пользователя. По заданной информации о пользователях и свободном объёме на архивном диске определите максимально возможное количество файлов, которые можно сохранить в архиве, а также максимальный размер имеющегося файла в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Входные данные. В первой строке входного файла `26.txt` находятся два числа: S – объём свободного пространства на диске (натуральное число, не превышающее 10 000) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов пользователей (натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы можно сохранить в архив, затем максимальный размер имеющегося файла, который может быть сохранен в архиве при условии, что сохранены файлы максимально возможного числа пользователей.

Пример входного файла:

```
100 4
80
30
50
40
```



```
26_ - Блокнот
Файл  Правка  Формат  Вид  Справка
8200 970
34
35
4
30
18
16
26
5
5
4
39
10
5
30
21
48
```

Схема решения (программирование):

- Открыть текстовый файл для чтения.
- Прочитать в первой строке два числа: **s** – размер архива, **n** – количество файлов
- Данные в следующих **n** строках прочитать в целочисленный массив/список **data** (значения объемов файлов пользователей). Элементы нумеруются с 0 до n-1.
- Массив/список упорядочить по возрастанию.
- Начиная с начала массива/списка, суммировать числа, пока текущая сумма **sm** не превышает **s** и подсчитывать количество файлов **cnt**, попавших в архив (начинаем с самых маленьких, чтобы взять максимальное количество файлов).
- Вычисляем запас **delta** (можно ли заменить максимальный файл в архиве на файл большего размера) и возможный размер файла для замены **rest** :

$$\mathbf{delta = s - sm}$$

$$\mathbf{rest = data[cnt - 1] + delta}$$

- Если **delta** > 0, выберем подходящие элементы для замены и определим максимальный из них.
- Проверим и выведем ответ.

```
1 with open('26.txt', 'r') as fin:
2     s, n = map(int, fin.readline().split())
3     data = fin.readlines()
4     data = sorted(map(int, data))
5     sm = cnt = 0
6     for i in range(n):
7         if sm + data[i] > s:
8             break
9         sm += data[i]
10        cnt += 1
11    print(cnt)
12    delta = s - sm
13    a = [elem for elem in data[cnt:] if elem-data[cnt-1] <= delta]
14    if len(a) != 0:
15        print(a[-1])
16    else:
17        print(data[cnt-1])
```

Debug I/O

Python Shell

Commands execute without debug. Use arrow keys for history.

3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)]

Python Type "help", "copyright", "credits" or "license" for more information.

>>> [evaluate demo26_2.py]

568

50

>>> |

Задание 26



Задание выполняется с использованием прилагаемых к заданию файлов.

Организация купила для своих сотрудников все места в нескольких рядах идущих рядах на концертной площадке. Известно, какие места уже распределены между сотрудниками. Найдите ряд с наибольшим номером, в котором есть два соседних места, таких что слева и справа от них в том же ряду места уже распределены (заняты). Гарантируется, что есть хотя бы один ряд, удовлетворяющий этому условию. В ответе запишите два целых числа: номер ряда и наименьший номер места из найденных в этом ряду подходящих пар свободных мест.

Входные данные

В первой строке входного файла находится число N – количество занятых мест (натуральное число, не превышающее 10 000). Каждая из следующих N строк содержит два натуральных числа, не превышающих 100 000: номер ряда и номер занятого места.

Выходные данные

Два целых неотрицательных числа: номер ряда и наименьший номер места в выбранной паре.

Пример входного файла:

```
7
40 3
40 6
60 33
50 125
50 128
50 64
50 67
```

301_26 – Блокнот

Файл Правка Формат

```
9986
13647 26316
27383 88746
63292 62533
3388 64244
1772 91844
64242 85129
9825 88362
25972 86056
16985 81310
67203 85818
29069 65141
28460 87334
27471 81815
61284 92165
33598 92485
```

Идея решения:

1. Заполняется двумерный массив n строк 2 столбца.
2. Массив сортируется при помощи лямбда-функции сначала по столбцу 2, потом по столбцу 1.
3. В цикле просматривается отсортированный двумерный массив. Переменная k будет хранить номер ряда в случае, когда у двух элементов ряд одинаковый и места с разницей 3, а переменная m будет хранить наименьший номер места из найденных в этом ряду подходящих пар свободных мест. При этом сохраняем ряд, если его номер больше текущего значения k . Ряд с таким же номером, но более поздние места не будут сохраняться, поскольку ряд меняется только, если находится гарантированно больший текущего.
4. Выводится ответ

Решение задачи на языке Python:

```
S = open('301_26.txt')
n = int(s.readline())
a = []
for i in range(n):
    temp = s.readline()
    a.append([int(j) for j in temp.split()])

a.sort(key=lambda x: x[1])
a.sort(key=lambda x: x[0])

k = 0
m = 10000
for i in range(n - 1):
    if a[i][0] == a[i+1][0] and a[i+1][1] - a[i][1] == 3 and k < a[i][0]:
        k = a[i][0]
        m = a[i][1] + 1
print(k, m)
```

- Программа
- Учебник 7-9
- Учебник 10-11
- Учебник 10-11
- Пособие (Ру, С++)
- Конкурсы
- Презентации
- ОГЭ (9 класс)

- ЕГЭ**
- Тесты (NetTest)
- Онлайн-тесты
- Генератор
- Тренажёр
- Программы
- Статьи
- Литература
- Видео
- Дополнительн
- Прошлые год
- Вопросы
- Отзывы
- Blockly
- Робототехника
- Arduino

Пробники ЕГЭ ↓

Математика, Физика, Информатика, Химия, **Русский**, Обществознание, Литература, История, Иностранные языки, География, Биология

1 октября 2021 В закладки Обсудить Жалоба

Тренажёр компьютерного ЕГЭ по информатике

Пробные работы ЕГЭ по информатике

Готовые варианты с ответами формата КЕГЭ-2022.

- Вариант 1
- Вариант 2
- Вариант 3
- Вариант 4
- Вариант 5
- Вариант 6
- Вариант 7
- Вариант 8
- Вариант 9
- Вариант 10
- Вариант 11
- Вариант 12
- Вариант 13
- Вариант 14
- Вариант 15
- Вариант 16

ПРОЕКТ С УЧАСТИЕМ РАЗРАБОТЧИКОВ КИМ ЕГЭ

ФИПИ ШКОЛЕ

2022

ЕГЭ

ЕДИННЫЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН

ИНФОРМАТИКА И ИКТ

ТИПОВЫЕ ЭКЗАМЕНАЦИОННЫЕ ВАРИАНТЫ

С. С. КРЫЛОВ, Т. В. ЧУРЮЖКА

НОВАЯ МОДЕЛЬ КИМ ЕГЭ

20 ВАРИАНТОВ

Пример задачи 26 (сайт Полякова К.Ю. <https://kpolyakov.spb.ru/>)

№ 26. (Е. Джобс) Робот складывает монеты в ящики. Задача робота заполнить как можно большее количество ящиков монетами в количестве 100 штук. Роботу по конвейеру поступают монеты в количестве K , где K может быть от 1 до 99 монет. Известно, что робот может высыпать в ящик монеты в количестве K . Корзина должна быть высыпана в ящик полностью. Необходимо определить, сколько ящиков можно заполнить монетами так, чтобы в каждом из них было ровно по 100 монет.

Входные данные представлены в файле `26-J1.txt`. В первой строке записано число N – количество ящиков, в каждой из последующих N строк число K – количество монет в каждой корзине.

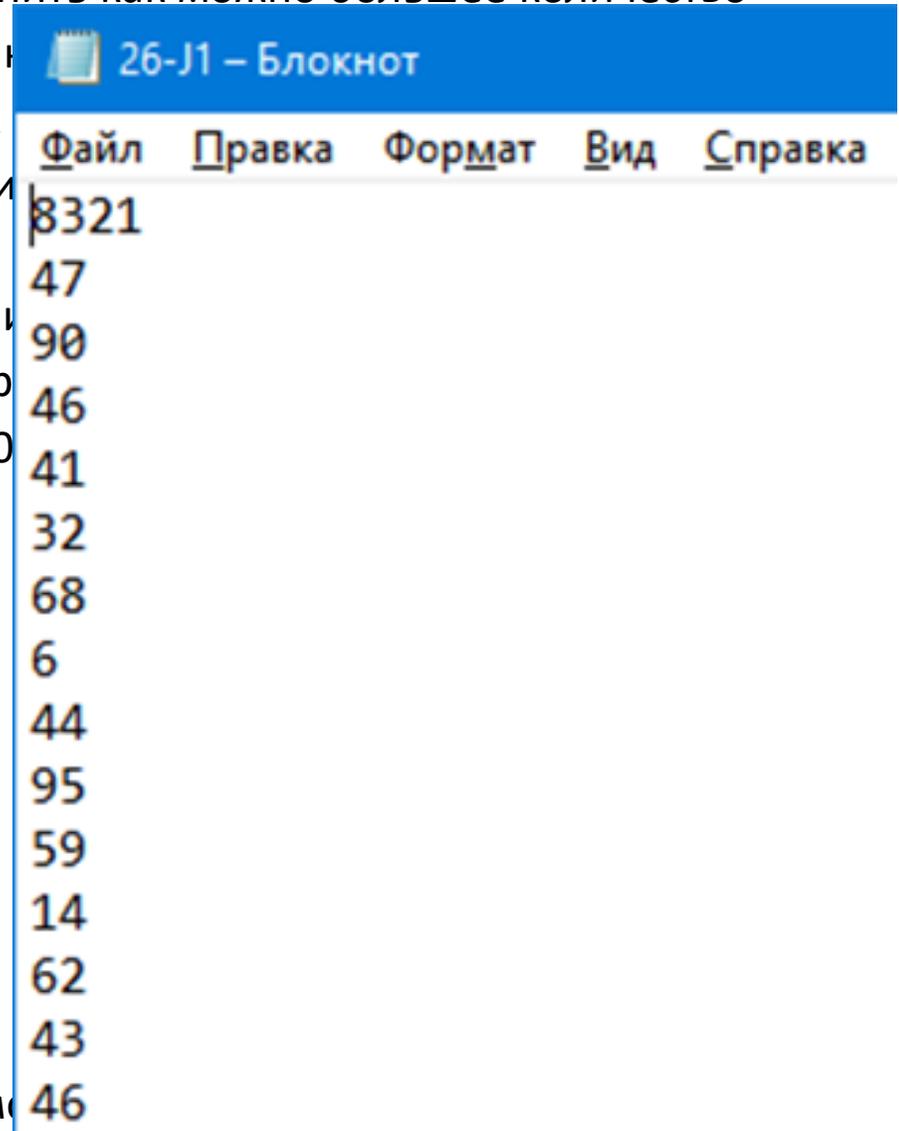
В качестве ответа дать одно число – количество ящиков, заполненных монетами.

Пример организации исходных данных во входном файле:

7
10
44
66
90
65
47
34

При таких исходных данных можно заполнить только 2 ящика по 100 монет.

Ответ: 2.



Идея решения:

- Количество монет в каждой корзине от 1 до 99, количество корзин N может быть очень большим.
- Нужно подсчитать, сколько раз встретилось каждое число от 1 до 99 во входных данных => общая сортировка данных ничем не поможет.
- Необходимо реализовать сортировку подсчетом. Для этого возьмем массив A из 99 элементов (от 1 до 99) и будем заполнять его в процессе чтения входных данных. Каждое прочитанное число X будет увеличивать на 1 элемент массива с индексом X .
- Количество ящиков можно подсчитать как количество пар элементов, у которых сумма равна 100: для каждого $a[i]$ парой будет $a[100-i]$ для всех i от 1 до 49 включительно. При этом если число 20, например, повторилось 35 раз, а число 80 повторилось только 15 раз, то к ответу нужно прибавить меньшее из этих двух значений.
- После цикла к ответу нужно добавить пары из $a[50]$, поделив их количество пополам, т.к. они образуют пары друг с другом.

```
1 fin = open('26-J1.txt', 'r')
2 n = int(fin.readline())
3 a = [0 for i in range(100)]
4 for i in range(n):
5     x = int(fin.readline())
6     a[x] += 1
7
8 ans = 0
9 for i in range(1, 50):
10     if a[i] != 0 and a[-i] != 0:
11         ans += min(a[i], a[-i])
12 ans += a[50] // 2
13
14 print(ans)
```

Debug I/O Python Shell

Commands execute without debug. Use arrow keys for history.

3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)]

Python Type "help", "copyright", "credits" or "license" for more information.

>>> [evaluate 26_Поляков.py]

3845

>>> |

Пример задачи 26 (Статград от 17.03.2021)

26 В текстовом файле записан набор натуральных чисел, не превышающих 10^9 . Гарантируется, что все числа различны. Необходимо определить, сколько в наборе таких пар чётных чисел, что их среднее арифметическое тоже присутствует в файле, и чему равно наибольшее из средних арифметических таких пар.

Входные данные

Первая строка входного файла содержит целое число N – общее количество чисел в наборе. Каждая из следующих N строк содержит одно число.

Пример входного файла

```
6
3
8
14
11
2
17
```

В данном случае есть две подходящие пары: 8 и 14 (среднее арифметическое 11), 14 и 2 (среднее арифметическое 8). В ответе надо записать числа 2 и 11.

В ответе запишите два целых числа: сначала количество пар, затем наибольшее среднее арифметическое.

```
5000
793526109
77410981
64762417
36931226
281553135
440635440
268917940
666856212
496603171
669925289
339031915
865540392
279970130
122029627
773210734
888153658
859413347
484577507
855242658
830235686
803771200
723433421
```

Идея решения 1 («в лоб»):

- Выбрать четные числа в доп. список и посчитать все возможные средние арифметические пар. Для этого потребуется порядка $O(n^2)$ операций. Потом нужно пройти по исходному списку и искать каждое такое среднее. Это еще n операций.

Решение работает очень долго.

Идея решения 2:

- Отсортировать исходный список чисел.
- Для всех четных пар получить список средних арифметических.
- Для каждого среднего арифметического организовать проверку наличия этого числа в исходном списке, запоминая эти числа в другом списке.
- После цикла вывести длину списка отобранных чисел и последний элемент как максимальное число.

Храним много «лишних» чисел

```
1 with open('26.txt') as f:
2     n = int(f.readline())
3     data = []
4     for i in range(n):
5         data.append(int(f.readline()))
6 data.sort()
7
8 averages = []
9 for i in range(n - 1):
10     for j in range(i + 1, n):
11         if data[i] % 2 == 0 and data[j] % 2 == 0:
12             s = data[i] + data[j]
13             averages.append(s // 2 )
14 averages.sort()
15
16 selected = []
17 i = 0
18 for av in averages:
19     while i < n and data[i] < av:
20         i += 1
21     if i < n and data[i] == av:
22         selected.append(av)
23
24 print(len(selected), selected[-1])
25
```

По

Найти:

Заменить:

\ < >

| | |

I/O Отладки Оболочка Python

Команды выполняются без отладки. Используйте клавиши со стрел

```
3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:3
Python Type "help", "copyright", "credits" or "]"
>>> [анализируем Статг2.py]
15 976339247
>>>
```

Идея решения 3:

- Отсортировать исходный список чисел.
- Организовать цикл для поиска четных пар, и поиск среднего арифметического сделать с помощью алгоритма бинарного поиска.

```
1 def bin(l, r, k, a):
2     while r > l + 1:
3         m = (l + r) // 2
4         if a[m] == k:
5             return True
6         if a[m] > k:
7             r = m
8         else:
9             l = m
10    return False
11
```

```
1 def bin(l, r, k, a):
2     while r > l + 1:
3         m = (l + r) // 2
4         if a[m] == k:
5             return True
6         if a[m] > k:
7             r = m
8         else:
9             l = m
10    return False
11
12 fin = open('26.txt')
13 n = int(fin.readline())
14 x = []
15 for i in range(n):
16     x.append(int(fin.readline()))
17 x.sort()
18 ans = mx = 0
19 for i in range(n - 1):
20     for j in range(i + 1, n):
21         if x[i] % 2 == 0 and x[j] % 2 == 0:
22             sr = (x[i] + x[j]) // 2
23             if bin(i, j, sr, x):
24                 ans += 1
25                 mx = max(mx, sr)
26 print(ans, mx)
```

```
3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [I
Python Type "help", "copyright", "credits" or "licen:
>>> [анализируем Статград170321_26.py]
15 976339247
>>> |
```

Рекомендации

Алгоритмизация и программирование – ключ к успешной сдаче ЕГЭ

Использовать ресурсы с автоматической системой проверки решений.



Поиск Название курса, автор или предмет

На любом языке ▾

С сертификатами

Бесплатные

Искать

Онлайн-курсы ▾

Новые курсы

Популярные курсы

Подборка от Stepik

Победители Stepik Awards 2021

"Поколение Python": курс для начинающих

Тимур  Гув, Школа BEEGEEK, Благотворительный фонд "Айкью Опшн", Руслан Чаниев



★ 5 👤 317K ⌚ 40 ч 🗨️

Бесплатно

Программирование на Python

Bioinformatics Institute



★ 4.7 👤 555K ⌚ 19 ч 🗨️

Бесплатно

Введение в программирование (C++)

Академия Яндекса, Высшая школа экономики (НИУ ВШЭ)



★ 4.8 👤 290K ⌚ 17 ч 🗨️

Бесплатно

Интерактивный тренажер по SQL

Дальневосточный федеральный университет



★ 5 👤 101K ⌚ 33 ч 🗨️

Бесплатно

"Поколение Python": курс для продвинутых

Тимур  Гув, Школа BEEGEEK, Благотворительный фонд "Айкью Опшн", Руслан Чаниев



★ 5 👤 43.1K ⌚ 68 ч 🗨️

Бесплатно

Веб-разработка для начинающих: HTML и CSS

ITC (vk.com/itc.digital)



★ 4.7 👤 157K ⌚ 3 ч 🗨️

Бесплатно



УЧИТЕ ПИТОН

Бесплатный курс по программированию с нуля.
Работает прямо в браузере.

```
запустить  выполнить пошагово 
34 print('две строки равны')
35 else:
36     print('так не бывает')
37
38 # списки (изменяемые последовательности)
39 houses = ['Ravenclaw', 'Hufflepuff', 'Gryffindor']
40 houses.append('Slytherin')
41
42 # цикл 'for'. отступы вновь важны!
43 for house in houses:
44     print('Ten points to', house, end='\n')
45
```

ЛЕГКО НАЧАТЬ

Питонтьютор работает из браузера.
Не парьтесь по поводу установки среды
разработки, интерпретатора и других сложных
штук.

ПО-РУССКИ

Ошибки в программах неизбежны.
Наш пошаговый отладчик переводит
сообщения об ошибках на русский язык
и объясняет их.

РАЗНООБРАЗНО

У нас больше сотни задач разной сложности.
Когда вы решите задачу, мы покажем вам
другие варианты решения, чтобы вы смогли
изучить разные подходы.

<https://informatics.msk.ru>

Информатикс

Поиск курса Применить 

Курсы

▼ Свернуть всё

Изучение языка программирования

- Ввод-вывод, оператор присваивания, арифметические операции  
- Условный оператор  
- Операторы цикла  
- Типы данных  
- Массивы  
- Функции и процедуры. Рекурсия  
- Учим python (в стадии разработки)  
- Особенности языков программирования  
- Составление тестов к задачам  

Структуры данных и алгоритмы

- Поиск и сортировка  
- Структуры данных  
- Алгоритмы на графах  

Что происходит

- Старая версия сайта доступна по ссылке <https://old.informatics.msk.ru> (в режиме readonly с 14.08)
- Если у вас что-то не работает, пишите в чат поддержки в телеграмм, общение на свободные темы идет в чате флудильне
- Для авторизации учителей, заливки новых задач нужно завести тикет

К задаче №

Рабочий стол

- Мои курсы
- Создать курс
- Управление группами
- Управление мониторинги

Tutorial для учителей

Основное меню

-  Каталог по темам и источникам
-  Мои посылки

О сайте

-  Команда проекта
-  Новые материалы
-  FAQ

Курсы

- ▶ Изучение языка пр
- ▶ Структуры данных
- ▶ Авторские курсы
- ▶ Учебно-тренирово
- ▶ Кружки и уроки
- ▶ Личные олимпиад
- ▶ Командные олимпи
- ▶ Олимпиады для на
- ▶ Подготовка к ЕГЭ п
- ▶ Архив

▶ Развернуть всё

Авторские курсы

[В начало](#) / [Курсы](#) / [Авторские курсы](#)

Категории курсов:

description

Поиск курса

Применить

[Задачи из учебника К.Ю. Полякова и Е.А. Еремина](#)

Teacher: [Константин Поляков](#)
Teacher: [Анна Георгиевна Тамаревская](#)

Задачи из учебника К.Ю. Полякова и Е.А. Еремина (Информатика 10-11 классы. Углублённый уровень. М.: Бином, 2013).

[Фирма "1С". "Алгоритмы. Олимпиадное программирование на языке Java для школьников"](#)

Teacher: [Илья Андреев](#)

Клуб программистов

[Д. П. Кириенко. Программирование на языке Python \(школа 179 г. Москвы\)](#)

[Е. В. Андреева. Программирование — это так просто, программирование — это так сложно.](#)

Задачи из учебника

[Ф. Меньшиков. Олимпиадные задачи по программированию](#)

Программирование на python: Задачи

В начало / Курсы / Авторские курсы / Программирование на python / Тема 7 / Задачи



N	Name	Sum	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Екатерина Саламатова	22	+ -1	++	+++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	+
2	Данил Артюшенко	21	+	++	+++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++
3	Екатерина Трофимова	20	+ -1	++	+++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	++++	+
4	Тимофей Случаев	17	+ -1	++	+++	-2	++	-1	+++	-1	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	-1	-1
5	Анжелика Антипо	14	+ -1	+ -1	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+
6	Таисия Жайворонская	9																										

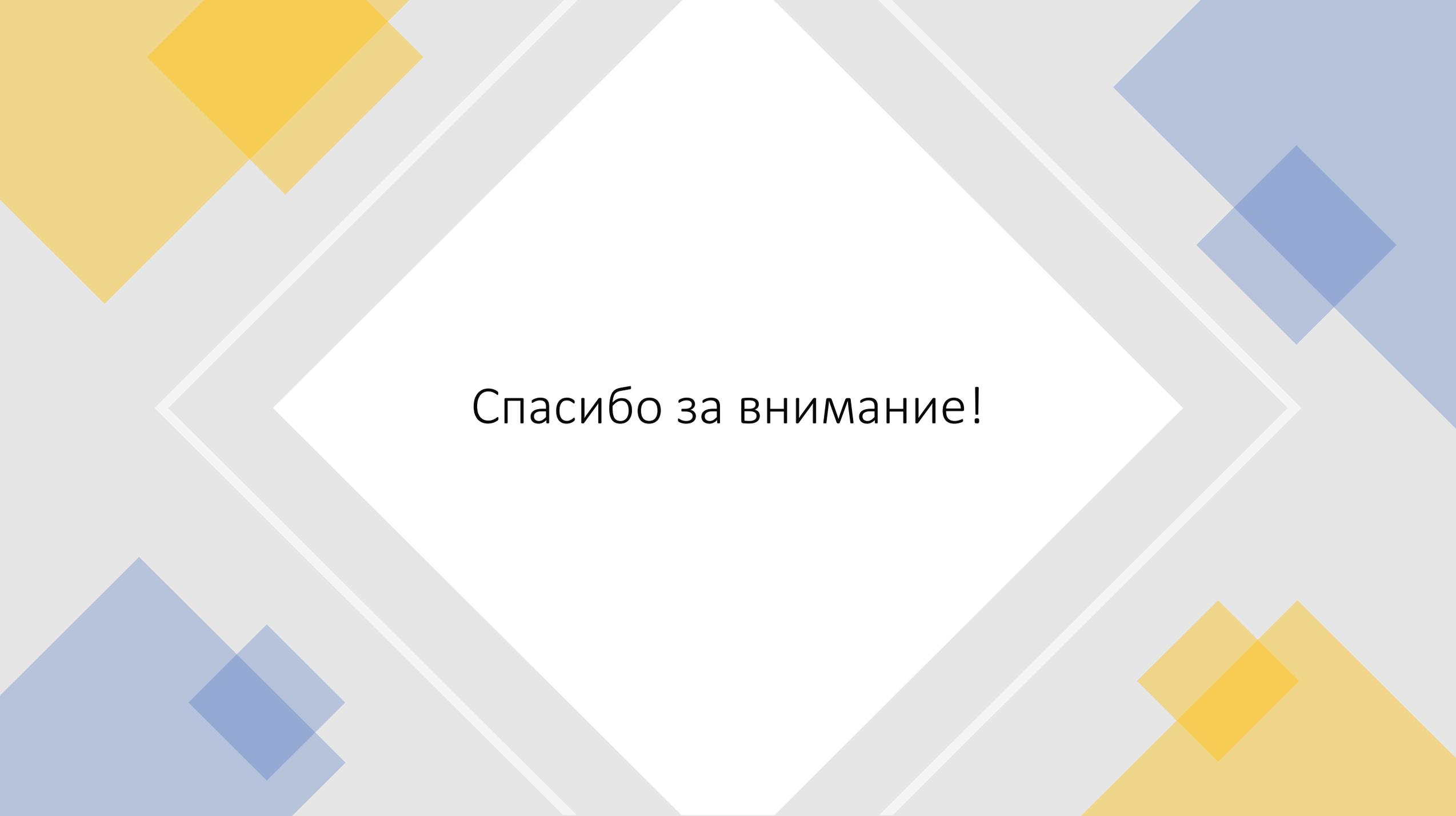
Letter	Name
Контекст	1660 Задачи
A	Список квадратов [3642]
B	Минимальный простой делитель [3643]
C	Список степеней двойки [3644]
D	Точная степень двойки [3645]
E	Двоичный логарифм [3646]
F	Утренняя пробежка [3647]
G	Банковские проценты [3648]
H	Длина последовательности [3649]
I	Сумма последовательности [3650]
J	Среднее значение последовательности [3651]
K	Среднее значение последовательности [3652]

[Все послыки](#)
[Результаты](#)

- Список групп
- [Все записавшиеся на курс](#)
 - группа 1
 - Сириус - Июнь 2018 (Отбор)
 - ege
 - 9 L
 - MOSH-2018-6
 - MOSH-2018-7
 - MOSH-2018-8
 - MOSH-2018-9
 - 8 La
 - 8 L
 - MOSH-2019-7
 - MOSH-2019-9
 - 6 L_DOP
 - 7 L
 - Подготовка
 - 7 LA

В заданиях КЕГЭ 2022 г. по сравнению с КЕГЭ 2021 г. и с демонстрационным вариантом 2022 г. **возможно** обновление сюжетов заданий без изменения уровня сложности, проверяемого элемента содержания и формы задания.





Спасибо за внимание!