

Алгоритмы, структуры данных, конструкции языка Python при подготовке к КЕГЭ по информатике

КЕГЭ 2021

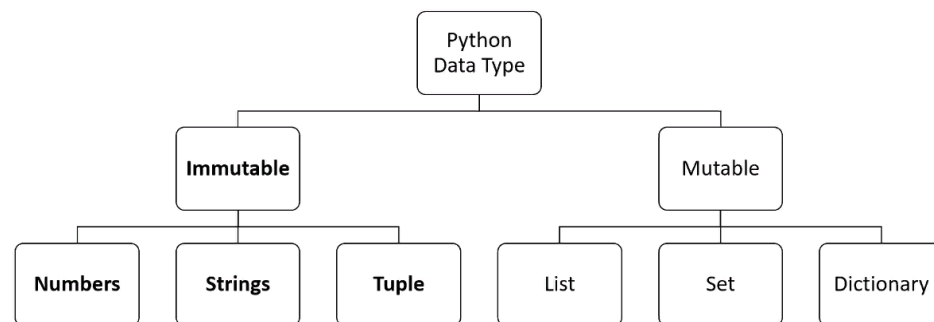
- Часть заданий убрали из спецификации
- Некоторые задания были переформулированы
- Добавлены новые задания
- Нумерация практически всех заданий изменилась
- На протяжении всего экзамена будет доступен компьютер
- Из 27 заданий только 9 требуют обязательного использования компьютера, но в остальных заданиях компьютер тоже доступен

Насколько компьютер полезен

- Для расчетов есть калькулятор
- Часть заданий (требующих и не требующих использования компьютера) может быть решена с использованием электронных таблиц или языка программирования

В каждом конкретном случае надо понимать: насколько целесообразно решать задание на компьютере, а не на бумаге.

Почему Python



Built-in Functions				
<code>abs()</code>	<code>divmod()</code>	<code>input()</code>	<code>open()</code>	<code>staticmethod()</code>
<code>all()</code>	<code>enumerate()</code>	<code>int()</code>	<code>ord()</code>	<code>str()</code>
<code>any()</code>	<code>eval()</code>	<code>isinstance()</code>	<code>pow()</code>	<code>sum()</code>
<code>basestring()</code>	<code>execfile()</code>	<code>issubclass()</code>	<code>print()</code>	<code>super()</code>
<code>bin()</code>	<code>file()</code>	<code>iter()</code>	<code>property()</code>	<code>tuple()</code>
<code>bool()</code>	<code>filter()</code>	<code>len()</code>	<code>range()</code>	<code>type()</code>
<code>bytearray()</code>	<code>float()</code>	<code>list()</code>	<code>raw_input()</code>	<code>unicchr()</code>
<code>callable()</code>	<code>format()</code>	<code>locals()</code>	<code>reduce()</code>	<code>unicode()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>long()</code>	<code>reload()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>map()</code>	<code>repr()</code>	<code>xrange()</code>
<code>cmp()</code>	<code>globals()</code>	<code>max()</code>	<code>reversed()</code>	<code>zip()</code>
<code>compile()</code>	<code>hasattr()</code>	<code>memoryview()</code>	<code>round()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hash()</code>	<code>min()</code>	<code>set()</code>	
<code>delattr()</code>	<code>help()</code>	<code>next()</code>	<code>setattr()</code>	
<code>dict()</code>	<code>hex()</code>	<code>object()</code>	<code>slice()</code>	
<code>dir()</code>	<code>id()</code>	<code>oct()</code>	<code>sorted()</code>	

Кодификатор

1.7	Языки программирования
1.7.1	Типы данных
1.7.2	Основные конструкции языка программирования. Система программирования
1.7.3	Основные этапы разработки программ. Разбиение задачи на подзадачи

- Нахождение минимума и максимума двух, трёх, четырёх данных чисел без использования массивов и циклов.
- Нахождение всех корней заданного квадратного уравнения.
- Запись натурального числа в позиционной системе с основанием, меньшим или равным 10. Обработка и преобразование такой записи числа.
- Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).
- Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту и т.д.).
- Заполнение элементов одномерного и двумерного массивов по заданным правилам.
- Операции с элементами массива. Линейный поиск элемента. Вставка и удаление элементов в массиве. Перестановка элементов данного массива в обратном порядке. Суммирование элементов массива. Проверка соответствия элементов массива некоторому условию.
- Нахождение второго по величине (второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.
- Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.
- Операции с элементами массива, отобранными по некоторому условию (например, нахождение минимального чётного элемента в массиве, нахождение количества и суммы всех чётных элементов в массиве).
- Сортировка массива.
- Слияние двух упорядоченных массивов в один без использования сортировки.
- Обработка отдельных символов данной строки. Подсчёт частоты появления символа в строке.
- Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

Задание 2. Таблицы истинности

```
print(' x ', ' y ', ' z ', ' w ', ' F ', sep='\t')
for x in (False, True):
    for y in (False, True):
        for z in (False, True):
            for w in (False, True):
                F = (x or y) and not (y == z) and not w
                if F:
                    print(x, y, z, w, F, sep='\t')
```

				$(x \vee y) \wedge \neg(y = z) \wedge \neg w$
1		1		1
0	1		0	1
	1	1	0	1

x	y	z	w	F
False	True	False	False	True
True	False	True	False	True
True	True	False	False	True

Задание 5. Алгоритм

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .
2. К этой записи дописываются справа ещё два разряда по следующему правилу:
 - а) складываются все цифры двоичной записи числа N , и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
 - б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы её цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите такое **наименьшее** число N , для которого результат работы данного алгоритма больше числа 77. В ответе это число запишите в десятичной системе счисления.

```
n = 1
res = 0
while res <= 77:
    res = bin(n)[2:]
    res += str(sum([int(i) for i in bin(n)[2:]] % 2))
    res += str(sum([int(i) for i in bin(n)[2:]] % 2))
    res = int(res, 2)
    print(n, res)
    n += 1
```

task_5 x

3	12
4	19
5	20
6	24
7	31
8	35
9	36
10	40
11	47
12	48
13	55
14	59
15	60
16	67
17	68
18	72
19	79

Задание 6. Программа

6

Определите, при каком наименьшем введённом значении переменной s программа выведет число 64. Для Вашего удобства программа представлена на четырёх языках программирования.

Паскаль	Python	Алгоритмический язык
<pre>var s, n: integer; begin readln (s); n := 1; while s < 51 do begin s := s + 5; n := n * 2 end; writeln(n) end.</pre>	<pre>s = int(input()) n = 1 while s < 51: s = s + 5 n = n * 2 print(n)</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n, s <u>ввод</u> s n := 1 <u>нц пока</u> s < 51 s := s + 5 n := n * 2 <u>кц</u> <u>вывод</u> n <u>кон</u></pre>

```
for s in range(100):
    print(s, end=' ')
n = 1
while s < 51:
    s += 5
    n *= 2
print(n)
```

```
0 2048
1 1024
2 1024
3 1024
4 1024
5 1024
6 512
7 512
8 512
9 512
10 512
11 256
12 256
13 256
14 256
15 256
16 128
17 128
18 128
19 128
20 128
21 64
22 64
23 64
24 64
25 64
```


Задание 12. Редактор

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 70 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (2222) ИЛИ нашлось (8888)

 ЕСЛИ нашлось (2222)

 ТО заменить (2222, 88)

 ИНАЧЕ заменить (8888, 22)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

```
s = '8' * 70
while '2222' in s or '8888' in s:
    if '2222' in s:
        s = s.replace('2222', '88')
    else:
        s = s.replace('8888', '22')
print(s)
```

```
C:\Python\Python3.8\python.exe
22
```

Задание 16. Рекурсивная функция

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$F(n) = 1$ при $n = 1$;

$F(n) = n + F(n - 1)$, если n – чётно,

$F(n) = 2 \times F(n - 2)$, если $n > 1$ и при этом n – нечётно.

Чему равно значение функции $F(26)$?

```
def f(n):  
    if n == 1:  
        return n  
    if n % 2 == 0:  
        return n + f(n - 1)  
    else:  
        return 2 * f(n - 2)  
  
print(f(26))
```

```
C:\Python\Python3.8\python.exe  
4122
```

Задание 17. Делимость чисел

Рассматривается множество целых чисел, принадлежащих числовому отрезку [1016; 7937], которые делятся на 3 и не делятся на 7, 17, 19, 27.

Найдите количество таких чисел и максимальное из них.

В ответе запишите два целых числа: сначала количество, затем максимальное число.

Для выполнения этого задания можно написать программу или воспользоваться редактором электронных таблиц.

```
count = 0
max_ = 0
for i in range(1016, 7938):
    if i % 3 == 0 and i % 7 != 0 and i % 17 != 0 and \
        i % 19 != 0 and i % 27 != 0:
        count += 1
        max_ = max(i, max_)
print(count, max_)
```

```
1 data = [i for i in range(1016, 7938) if i % 3 == 0 and
2     all(i % x != 0 for x in (7, 17, 19, 27))]
3 print(len(data))
4 print(data[-1])
5
```

Задание 22. Циклический алгоритм

Ниже на четырёх языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: L и M . Укажите **наибольшее** число x , при вводе которого алгоритм печатает сначала 4, а потом 5.

C++

```
#include <iostream>
using namespace std;

int main()
{
    int x, L, M, Q;
    cin >> x;
    Q = 9;
    L = 0;
    while (x >= Q) {
        L = L + 1;
        x = x - Q;
    }
    M = x;
    if (M < L) {
        M = L;
        L = x;
    }
    cout <<L <<endl <<M <<endl;
    return 0;
}
```

Python

```
x = int(input())
Q = 9
L = 0
while x >= Q:
    L = L + 1
    x = x - Q
M = x
if M < L:
    M = L
    L = x
print(L)
print(M)
```

```
for x in range(100000):
```

```
    x_ = x
    Q = 9
    L = 0
    while x >= Q:
        L = L + 1
        x = x - Q
    M = x
    if M < L:
        M = L
        L = x
    if L == 4 and M == 5:
        print(x_, L, M)
```

```
C:\Python\Python3.8\python.exe
```

```
41 4 5
```

```
49 4 5
```

Задание 24. Обработка строк

Текстовый файл состоит не более чем из 10^6 символов X, Y и Z.

Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

Для выполнения этого задания следует написать программу.

```
s = open('24.txt').read().strip()
max_ = 0
i = 0
current = 1
while i < len(s) - 1:
    if s[i] != s[i + 1]:
        current += 1
    else:
        max_ = max(max_, current)
        current = 1
    i += 1
print(max_)
```

Задание 25. Обработка чисел

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.

```
1 for x in range(174457, 174506):
2     div = []
3     for i in range(2, x // 2 + 1):
4         if x % i == 0:
5             div.append(i)
6     if len(div) == 2:
7         print(i, div)
8
```

```
C:\Python\Python3.8\python.exe
87229 [3, 58153]
87230 [7, 24923]
87231 [59, 2957]
87236 [13, 13421]
87239 [149, 1171]
87242 [5, 34897]
87248 [211, 827]
87251 [2, 87251]
```

Задание 26. Обработка чисел

Входные данные.

В первой строке входного файла находятся два числа: S – размер свободного места 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 каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

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

Пример входного файла:

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

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар – 50, поэтому ответ для приведённого примера:

```
1 with open('26.txt') as f:
2     data = list(map(str.strip, f.readlines()))
3     s, n = map(int, data.pop(0).split())
4     users = list(map(int, data))
5     users.sort()
6     max_users = 0
7     for i in range(1, n):
8         current = sum(users[:i])
9         count = i
10        if s >= current and count >= max_users:
11            max_users = count
12            sum_ = current
13            max_user = users[:i][-1]
14    big = 0
15    for i in range(users.index(max_user), n):
16        if users[i] - max_user <= s - sum_:
17            big = max(big, users[i])
18    print(max_users, big)
```

Задание 27. Обработка последовательности чисел

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.

1	3
2	1 2
3	3 4
4	5 6
5	

20
5627 5841
5544 6520
1449 3580
2984 5984
6164 2583

60000
7722 7518
906 1474
859 1688
425 3358
2312 8232
5322 1618

Задание 27. Обработка последовательности чисел

```
1 from time import time
2
3 start = time()
4 with open('27-A.txt') as f:
5     data = list(map(str.strip, f.readlines()))
6 n = int(data.pop(0))
7 s = [0]
8 for i in range(n):
9     print(i)
10    print('time:', round(time() - start), 's')
11    a, b = map(int, data[i].split())
12    for j in range(len(s)):
13        x = s.pop(0)
14        s.append(x + a)
15        s.append(x + b)
16 s.sort()
17 while s and s[-1] % 3 == 0:
18     s.pop()
19 print(s[-1])
20 print('time:', round(time() - start), 's')
```

yandex-disk	andrew	0	2237	94,9 MiB
python3	andrew	25	10915	42,7 MiB
jetbrains-toolb	andrew	0	3617	33,9 MiB

```
13|
time: 0 s
14
time: 0 s
15
time: 0 s
16
time: 1 s
17
time: 5 s
18
time: 21 s
19
time: 106 s
127127
time: 896 s
```

Задание 27. Обработка последовательности чисел

```
1 with open('27-B.txt') as f:
2     data = list(map(str.strip, f.readlines()))
3     n = int(data.pop(0))
4     s = 0
5     min_ = 10000
6     for i in range(n):
7         x1, x2 = map(int, data[i].split())
8         s += max(x1, x2)
9         delta = max(x1, x2) - min(x1, x2)
10        if delta % 3 != 0:
11            min_ = min(delta, min_)
12    if s % 3 == 0:
13        s -= min_
14    print(s)
```

В итоге

Структуры данных: в демо-версии 2021 только списки.

Рекомендация: множества, словари.

Методы списков: в демо-версии: срезы, `min()`, `max()`, `sum()`, `sort()`

Методы строк: в демо-версии 2021 `len()`, `replace()`, `find()`, `in`, `strip()`

Рекомендация: срезы, `split()`, `join()`

Работа с файлами: чтение текстовых файлов различными способами

Дополнительно: `map`, списочные выражения, `bin oct hex int`,

Алгоритмы, структуры данных, конструкции языка Python при подготовке к КЕГЭ по информатике