

# **ЕГЭ по информатике 2022**

учитель информатики МАОУ СОШ № 88 г. Тюмени

Басс Ирина Сергеевна

# Характеристика структуры и содержания КИМ ЕГЭ по информатике

*Распределение заданий по уровням сложности*

Уровень сложности заданий	Количество заданий	Максимальный первичный балл	Процент максимального первичного балла за выполнение заданий данного уровня сложности от максимального первичного балла за всю работу
Базовый	11	11	38
Повышенный	11	11	38
Высокий	5	7	24
Итого	27	29	100

# Характеристика структуры и содержания КИМ ЕГЭ по информатике Задания базового уровня сложности

Номер задания	Элементы содержания
1	Представление и считывание данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)
2	Таблицы истинности и логические схемы.
3	Поиск информации в реляционных базах данных.
4	Кодирование и декодирование информации.
5	Формальное исполнение простого алгоритма, записанного на естественном языке, или создание линейного алгоритма для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы.
6	Знание основных конструкций языка программирования, понятия переменной, оператора присваивания.
7	Определение объема памяти, необходимого для хранения графической и звуковой информации.
8	Основные понятия и методы , используемые при измерении количества информации.
9	Умение обрабатывать числовую информацию в электронных таблицах.
10	Информационный поиск средствами операционной системы или текстового процессора.
19	Умение анализировать алгоритм логической игры.

# Характеристика структуры и содержания КИМ ЕГЭ по информатике

## Задания повышенного уровня сложности

Номер задания	Элементы содержания
11	Информационный объем сообщения.
12	Алгоритм для конкретного исполнителя с фиксированным набором команд.
13	Представление и считывание данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы).
14	Позиционные системы счисления.
15	Основные понятия и законы математической логики.
16	Вычисление рекуррентных выражений.
17	Алгоритм обработки числовой последовательности.
18	Обработка числовых данных в электронных таблицах.
20	Поиск выигрышной стратегии игры.
22	Анализ алгоритма, содержащего ветвления и циклы.
23	Анализ результата алгоритма, содержащего ветвление и цикл.

# Характеристика структуры и содержания КИМ ЕГЭ по информатике **Задания высокого уровня сложности**

Номер задания	Элементы содержания
21	Построение дерева игры по заданному алгоритму и поиск выигрышной стратегии.
24	Программа для обработки символьной информации.
25	Программа для обработки целочисленной информации.
26	Обработка целочисленной информации с помощью сортировки.
27	Программа для анализа числовых последовательностей.

# Обновленные линии ЕГЭ 2022 г. (по проекту демо 2022 г.)

---

**ЛИНИЯ 3** ➤ реляционные базы данных:  
от традиционной формы в компьютерную

---

**ЛИНИЯ 9** ➤ обработка числовых данных в электронных таблицах:  
от обработки диапазонов к усложненным логическим условиям

---

**ЛИНИЯ 17** ➤ программирование обработки целых чисел:  
от обработки диапазона чисел к обработке массивов данных

---

## ЗАДАНИЕ 2

---

Миша заполнял таблицу истинности логической функции  $F$

$$\neg(y \rightarrow (x \equiv w)) \wedge (z \rightarrow x),$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

				$F$
	1	1		1
0			0	1
	0	1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

## ЗАДАНИЕ 2

---

```
1 print('x y z w')
2
3 for x in range(2):
4     for y in range(2):
5         for z in range(2):
6             for w in range(2):
7                 F = (not (y <= (x == w))) and (z <= x)
8                 if F == 1:
9                     print(x, y, z, w)
```

**ВЫВОД:**

x	y	z	w
0	1	0	1
1	1	0	0
1	1	1	0



## ЗАДАНИЕ 2

---

```
1 print('x y z w')
2
3 for x in [0, 1]:
4     for y in [0, 1]:
5         for z in [0, 1]:
6             for w in [0, 1]:
7                 F = (not (y <= (x == w))) & (z <= x)
8                 if F:
9                     print(x, y, z, w)
```

**ВЫВОД:**

x	y	z	w
0	1	0	1
1	1	0	0
1	1	1	0













# ЗАДАНИЕ 2

---

	A	B	C	D	E	F	G	H	I
1	x	y	z	w					
2	0	1	0	1					
3	1	1	0	0					
4	1	1	1	0					
5									
6	w	x	y	z					
7	0	1	1	1					
8	0	1	1	0					
9	1	0	1	0					
10									
11									
12									
13									

ОТВЕТ: wxyz



## ЗАДАНИЕ 5

---

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 результирующего 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.

# ЗАДАНИЕ 5

---

```
1  for n in range(1, 100):
2      n2 = bin(n)  # 0b1100011, <class 'str'>
3      n2 = n2[2:]  # убираем символы '0b'
4      if n2.count('1') % 2 == 1:  # считаем сумму единиц
5          n2 = n2 + '1'
6      else:
7          n2 = n2 + '0'
8
9      if n2.count('1') % 2 == 1:
10         n2 = n2 + '1'
11     else:
12         n2 = n2 + '0'
13
14     # переводим строковое представление числа
15     # из двоичной записи в десятичную
16     r = int(n2, 2)
17     print(n, r)
```

# ЗАДАНИЕ 5

```
1 for n in range(1, 100):
2     n2 = bin(n) # 0b1100011, <class 'str'>
3     n2 = n2[2:] # убираем символы '0b'
4     if n2.count('1') % 2 == 1: # считаем сумму единиц
5         n2 = n2 + '1'
6     else:
7         n2 = n2 + '0'
8
9     if n2.count('1') % 2 == 1:
10        n2 = n2 + '1'
11    else:
12        n2 = n2 + '0'
13
14    r = int(n2, 2)
15    if r > 77:
16        print(n, r)
```

ВЫВОД: 19 78

20 80

21 86

22 90

23 92

24 96

25 102

26 106

27 108

28 114

29 116

ОТВЕТ: 19

# ЗАДАНИЕ 6

Определите, при каком наибольшем 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);   s := s div 10;   n := 1;   while s &lt; 51 do   begin     s := s + 5;     n := n * 2   end;   writeln(n) end.</pre>	<pre>s = int(input()) s = s // 10 n = 1 while s &lt; 51:     s = s + 5     n = n * 2 print(n)</pre>	<pre>алг нач   цел n, s   ввод s   s := div(s, 10)   n := 1   нц пока s &lt; 51     s := s + 5     n := n * 2   кц   вывод n кон</pre>
<b>C++</b>		
<pre>#include &lt;iostream&gt; using namespace std;  int main() {   int s, n;   cin &gt;&gt; s;   s = s / 10;   n = 1;   while (s &lt; 51) {s = s + 5; n = n * 2;}   cout &lt;&lt; n &lt;&lt; endl;   return 0; }</pre>		

## ЗАДАНИЕ 6

---

```
1  for i in range(1, 1000):  
2      s = i  
3      s = s // 10  
4      n = 1  
5      while s < 51:  
6          s = s + 5  
7          n = n * 2  
8      print(i, n)
```

**ВЫВОД:**

253	64
254	64
255	64
256	64
257	64
258	64
259	64
260	32
261	32
262	32

## ЗАДАНИЕ 6

---

```
1  for i in range(1, 1000):
2      s = i
3      s = s // 10
4      n = 1
5      while s < 51:
6          s = s + 5
7          n = n * 2
8      if n == 64:
9          print(i)
```

ВЫВОД:

252

253

254

255

256

257

258

259

Process finished with exit code 0

ОТВЕТ: 259

## ЗАДАНИЕ 8

---

Все четырёхбуквенные слова, в составе которых могут быть только буквы Л, Е, М, У, Р, записаны в алфавитном порядке и пронумерованы, начиная с 1.

Ниже приведено начало списка.

1. ЕЕЕЕ
2. ЕЕЕЛ
3. ЕЕЕМ
4. ЕЕЕР
5. ЕЕЕУ
6. ЕЕЛЕ
- ...

Под каким номером в списке идёт первое слово, которое начинается с буквы Л?

## ЗАДАНИЕ 8

---

```
1 s = 'ЕЛМРУ'  
2 num = 1  
3 for b1 in s:  
4     for b2 in s:  
5         for b3 in s:  
6             for b4 in s:  
7                 word = b1 + b2 + b3 + b4  
8                 print(num, word)  
9                 num = num + 1
```

ВЫВОД:

124	ЕУУР
125	ЕУУУ
126	ЛЕЕЕ
127	ЛЕЕЛ
128	ЛЕЕМ
129	ЛЕЕР
130	ЛЕЕУ

ОТВЕТ: 126



# ЗАДАНИЕ 8

---

```
1 s = 'ЕЛМРУ'  
2 num = 1  
3 for b1 in s:  
4     for b2 in s:  
5         for b3 in s:  
6             for b4 in s:  
7                 word = b1 + b2 + b3 + b4  
8                 if word[0] == 'Л':  
9                     print(num, word)  
10                num = num + 1
```

ВЫВОД:

```
126 ЛЕЕЕ  
127 ЛЕЕЛ  
128 ЛЕЕМ  
129 ЛЕЕР  
130 ЛЕЕУ  
131 ЛЕЛЕ  
132 ЛЕЛЛ
```

ОТВЕТ: 126

## ЗАДАНИЕ 12

---

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**

# ЗАДАНИЕ 12

---

```
1 s = '8' * 70
2
3 while '2222' in s or '8888' in s:
4     if '2222' in s:
5         s = s.replace('2222', '88', 1)
6     else:
7         s = s.replace('8888', '22', 1)
8
9 print(s)
10
```

ОТВЕТ: 22

## ЗАДАНИЕ 14

---

Значение арифметического выражения

$$3 \cdot 4^{38} + 2 \cdot 4^{23} + 4^{20} + 3 \cdot 4^5 + 2 \cdot 4^4 + 1$$

записали в системе счисления с основанием 16.

Сколько значащих нулей содержится в этой записи?

## ЗАДАНИЕ 14

---

```
1  x = 3*4**38 + 2*4**23 + 4**20 + 3*4**5 + 2*4**4 + 1
2  k = 0 # счетчик нулей
3  while x > 0:
4      if x % 16 == 0:
5          k = k + 1
6      x = x // 16
7
8  print(k)
```

ОТВЕТ: 15

## ЗАДАНИЕ 16

---

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

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n + F(n - 1), \text{ если } n \text{ чётно,}$$

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

Чему равно значение функции  $F(26)$ ?

## ЗАДАНИЕ 16

---

```
1  def f(n):
2      if n == 1:
3          return 1
4      if n % 2 == 0:
5          return n + f(n - 1)
6      if n > 1 and n % 2 != 0:
7          return 2 * f(n - 2)
8
9
10 print(f(26))
```

ОТВЕТ: 4122

## ЗАДАНИЕ 17

---

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от  $-10\,000$  до  $10\,000$  включительно. Определите и запишите в ответе сначала количество пар элементов последовательности, в которых хотя бы одно число делится на 3, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Например, для последовательности из пяти элементов: 6; 2; 9; -3; 6 – ответ: 4 11



## ЗАДАНИЕ 17

---

```
1 with open('17.txt') as f:  
2     s = f.read().split('\n')  
3 print(s)
```

['-67', '-694', '835', '786', '-457', '495', '-699', '-592', '-405', '-745', '656', '501',

```
1 with open('17.txt') as f:  
2     s = [int(x) for x in f]  
3 print(s)
```

[-67, -694, 835, 786, -457, 495, -699, -592, -405, -745, 656, 501, 48, 856,

# ЗАДАНИЕ 17

---

```
1 with open('17.txt') as f:
2     s = [int(x) for x in f]
3
4     mx = 0
5     k = 0
6     for i in range(0, len(s) - 1):
7         if s[i] % 3 == 0 or s[i + 1] % 3 == 0:
8             k = k + 1
9             if s[i] + s[i + 1] > mx:
10                mx = s[i] + s[i + 1]
11
12     print(k, mx)
```

ОТВЕТ: 2802 1990

## ЗАДАНИЕ 19-21

---

```
1 print(all([1, 1, 1])) # True
2 print(all([1, 0, 1])) # False
3 print(any([0, 1, 0])) # True
4 print(any([0, 0, 0])) # False
5
6 print([x ** 2 for x in (1, 2, 3, 4, 5)])
7 # [1, 4, 9, 16, 25]
8
9 print([2 * x for x in range(8)])
10 # [0, 2, 4, 6, 8, 10, 12, 14]
```

## ЗАДАНИЕ 19 (К.Ю. Поляков)

---

(№ 2416) Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **три камня** или **увеличить** количество камней в куче **в два раза**. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 75. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 75 или больше камней.

В начальный момент в первой куче было 9 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 65$ . Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Ответьте на следующие вопросы:

**Вопрос 1.** Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение  $S$ , при котором это возможно.

## ЗАДАНИЕ 19-21

---

```
3  def move(x, y):  
4  return (x + 3, y), (x * 2, y), (x, y + 3), (x, y * 2)
```

## ЗАДАНИЕ 19-21

---

```
3  def move(x, y):  
4      return (x + 3, y), (x * 2, y), (x, y + 3), (x, y * 2)
```

```
7  def game(x, y):  
8      if x + y >= 75:  
9          return 'Win'  
10     if any(game(a, b) == 'Win' for a, b in move(x, y)):  
11         return 'V1'  
12     if all(game(a, b) == 'V1' for a, b in move(x, y)):  
13         return 'P1'  
14     if any(game(a, b) == 'P1' for a, b in move(x, y)):  
15         return 'V2'  
16     if all(game(a, b) == 'V1' or game(a, b) == 'V2' for a, b in move(x, y)):  
17         return 'P2'
```

# ЗАДАНИЕ 19-21

```
1 from functools import lru_cache
2
3 def move(x, y):
4     return (x + 3, y), (x * 2, y), (x, y + 3), (x, y * 2)
5
6 @lru_cache(None)
7 def game(x, y):
8     if x + y >= 75:
9         return 'Win'
10    if any(game(a, b) == 'Win' for a, b in move(x, y)):
11        return 'V1'
12    if all(game(a, b) == 'V1' for a, b in move(x, y)):
13        return 'P1'
14    if any(game(a, b) == 'P1' for a, b in move(x, y)):
15        return 'V2'
16    if all(game(a, b) == 'V1' or game(a, b) == 'V2' for a, b in move(x, y)):
17        return 'P2'
```

## ЗАДАНИЕ 19-21

---

```
19 for s in range(1, 66):  
20     print(s, game(9, s))
```



# ЗАДАНИЕ 19-21

---

```
1  from functools import lru_cache
2
3  def move(x, y):
4      return (x + 3, y), (x * 2, y), (x, y + 3), (x, y * 2)
5
6  @lru_cache(None)
7  def game(x, y):
8      if x + y >= 75:
9          return 'Win'
10     if any(game(a, b) == 'Win' for a, b in move(x, y)):
11         return 'V1'
12     if all(game(a, b) == 'V1' for a, b in move(x, y)):
13         return 'P1'
14     if any(game(a, b) == 'P1' for a, b in move(x, y)):
15         return 'V2'
16     if all(game(a, b) == 'V1' or game(a, b) == 'V2' for a, b in move(x, y)):
17         return 'P2'
18
19  for s in range(1, 66):
20     print(s, game(9, s))
```

# ЗАДАНИЕ 19-21

---

1	None	11	None	21	None	31	V2	41	V1	51	V1	61	V1
2	None	12	None	22	None	32	P1	42	V1	52	V1	62	V1
3	None	13	None	23	None	33	V1	43	V1	53	V1	63	V1
4	None	14	None	24	P2	34	V1	44	V1	54	V1	64	V1
5	None	15	None	25	P2	35	V1	45	V1	55	V1	65	V1
6	None	16	V2	26	None	36	V1	46	V1	56	V1		
7	None	17	None	27	V2	37	V1	47	V1	57	V1		
8	None	18	None	28	V2	38	V1	48	V1	58	V1		
9	None	19	None	29	V2	39	V1	49	V1	59	V1		
10	None	20	None	30	V2	40	V1	50	V1	60	V1		

## ЗАДАНИЕ 20 (К.Ю. Поляков)

---

**Вопрос 2.** Укажите минимальное значение  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

14 None

15 None

16 V2

17 None

18 None

19 None

ОТВЕТ: 16

## ЗАДАНИЕ 21 (К.Ю. Поляков)

---

**Вопрос 3.** Найдите два значения  $S$ , при которых у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Найденные значения запишите в ответе в порядке возрастания.

21 None

22 None

23 None

24 P2

25 P2

26 None

27 V2

ОТВЕТ: 24 25

## ЗАДАНИЕ 19 (К.Ю. Поляков)

---

(№ 2416) Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **три камня** или **увеличить** количество камней в куче **в два раза**. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 75. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 75 или больше камней.

В начальный момент в первой куче было 9 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 65$ . Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Ответьте на следующие вопросы:

**Вопрос 1.** Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение  $S$ , при котором это возможно.

# ЗАДАНИЕ 19-21

```
6 @lru_cache(None)
7 def game(x, y):
8     if x + y >= 75:
9         return 'Win'
10    if any(game(a, b) == 'Win' for a, b in move(x, y)):
11        return 'V1'
12    if any(game(a, b) == 'V1' for a, b in move(x, y)):
13        return 'P1'
14    if any(game(a, b) == 'P1' for a, b in move(x, y)):
15        return 'V2'
16    if all(game(a, b) == 'V1' or game(a, b) == 'V2' for a, b in move(x, y)):
17        return 'P2'
```

ВЫВОД:

```
15 V2
16 V2
17 P1
18 P1
19 P1
```

ОТВЕТ: 17

# ЗАДАНИЕ 22

---

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++	Python
<pre>#include &lt;iostream&gt; using namespace std;  int main() {     int x, L, M, Q;     cin &gt;&gt; x;     Q = 9;     L = 0;     while (x &gt;= Q) {         L = L + 1;         x = x - Q;     }     M = x;     if (M &lt; L) {         M = L;         L = x;     }     cout &lt;&lt;L &lt;&lt;endl &lt;&lt;M &lt;&lt;endl;     return 0; }</pre>	<pre>x = int(input()) Q = 9 L = 0 while x &gt;= Q:     L = L + 1     x = x - Q M = x if M &lt; L:     M = L     L = x print(L) print(M)</pre>

## ЗАДАНИЕ 22

---

```
1   for i in range(1, 10000):
2       x = i
3       Q = 9
4       L = 0
5       while x >= Q:
6           L = L + 1
7           x = x - Q
8       M = x
9       if M < L:
10          M = L
11          L = x
12      if L == 4 and M == 5:
13          print(i)
```

ОТВЕТ: 49



# ЗАДАНИЕ 23

---

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 20, и при этом траектория вычислений содержит число 10?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

## ЗАДАНИЕ 22

---

```
1  def f(n, end):
2      if n > end:
3          return 0
4      if n == end:
5          return 1
6      if n < end:
7          return f(n + 1, end) + f(n * 2, end)
8
9
10 print(f(1, 10) * f(10, 20))
```

ОТВЕТ: 28

**СПАСИБО ЗА ВНИМАНИЕ!**