

# **Мониторинг готовности к ЕГЭ**

20 мая 2020 г.

# Мониторинг готовности к ЕГЭ (задания 24-27) –

20.05.2020

- 9.50 - Задания размещаются на сайт <http://www.pervayakyzyl.ru/>  
(в разделе Дистанционное обучение – Информатика)
- решения обязательны!
- разрешается использовать Pascal!!! (***файлы приложить к ответам***)
- сканирование всей работы
- ~~12.10-12.20~~ – отправка – . . . . .  
|            **Задания №24, 25, 26 даны в двух**            |  
|            **вариантах.**            |  
|            **Засчитывается лучший результат по**            |

**баллам**  
Осталось только две недели для консультаций.  
Присылайте заранее вопросы и № телефона по  
почте.

**№24-1.** Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится произведение цифр этого числа. Программист торопился и написал программу неправильно:

```
var N, product: longint;  
    digit: integer;  
begin  
    readln(N);  
    product := N mod 10;  
    while N >= 10 do begin  
        digit := N mod 10;  
        product := product * digit;  
        N := N div 10;  
    end;  
    writeln (product);  
end.
```

Последовательно выполните следующее:

1. Определите, что выведет программа при вводе числа 532 (обязательно запишите трассировку, т.е. процесс пошагового выполнения программы).
2. Приведите пример такого трёхзначного числа, при вводе которого приведённая программа, несмотря на ошибки, выдаёт верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - а) выпишите строку, в которой сделана ошибка;
  - б) укажите, как надо исправить ошибку - приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, имеющую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

**№24-2.** Дано натуральное число  $A$ . Требуется вывести такое минимально возможное натуральное число  $K$ , при котором сумма  $1 + (1 + 2) + (1 + 2 + 3) + \dots + (1 + 2 + \dots + K)$  окажется больше  $A$ . Стажёр торопился и написал программу неправильно.

```
var a, s, p, k: integer;
```

```
begin
```

```
  read(a);
```

```
  s := 0;
```

```
  p := 0;
```

```
  k := 1;
```

```
  while s < a do begin
```

```
    p := p+k;
```

```
    s := s+p;
```

```
    k := k+1;
```

```
  end;
```

```
  writeln(k);
```

```
end.
```

Последовательно выполните следующее:

1. Для каких из этих чисел 3, 56, 62, 84, 92 программа работает неправильно. Для каждого такого числа укажите верный ответ, а также ответ программы.
2. Найдите в программе все ошибки (известно, что их не более двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

**№25-1.** Дан целочисленный массив из 10 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит самую большую сумму двух соседних элементов и заменяет на эту сумму все элементы, кратные 3. Гарантируется, что в массиве есть хотя бы один элемент, кратный 3. В качестве результата необходимо вывести измененный массив в обратном порядке, каждый элемент массива выводится с новой строки. Например, для массива из восьми элементов:

**1 2 6 3 2 7 3 4**

программа должна вывести следующий массив:

**4 10 7 2 10 10 2 1**

Исходные данные объявлены так, как показано ниже на примере. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

```
const N = 10;  
var a: array [1..N] of longint;  
i, j, k: longint;  
begin  
    for i := 1 to N do  
        readln(a[i]);  
  
    ...  
end.
```

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия.

**№25-2.** Дан целочисленный массив из 10 элементов. Элементы массива могут принимать целые значения от -10 000 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит минимальный из элементов массива, которые двузначны и не кратны 11, а затем заменяет элементы, которые не оканчиваются на 3 на число, равное найденному значению. Если таких элементов нет, то считать минимум равным 0. В качестве результата необходимо вывести изменённый массив в обратном порядке, каждый элемент выводится с новой строки. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

```
const n=10;  
var a: array [1..n] of integer;  
    i, m, k: integer;  
begin  
for i:=1 to n do  
    readln(a[i]);  
...  
end.
```

**№26-1.** 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 (по своему выбору) один камень или увеличить количество камней в куче в три раза.** Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 38. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую позицию, что в кучах всего будет 38 или больше камней.

В начальный момент в первой куче было 8 камней, во второй куче –  $S$  камней;  $1 \leq S \leq 29$ .

*При каких  $S$ :*

- 1а) Петя выигрывает первым ходом; 1б) Ваня выигрывает первым ходом?*
- 2) Назовите значение  $S$ , при котором Петя может выиграть своим вторым ходом.*
- 3) Назовите значение  $S$ , при котором Ваня выигрывает своим первым или вторым ходом. Для указанного значения  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы).*

В узлах дерева указывайте позиции, на рёбрах рекомендуется указывать ходы. Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

**№26-2.** 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 два камня или увеличить количество камней в куче в четыре раза.**

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 61 или больше камней. В начальный момент в первой куче было 8 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 52$ .

Выполните следующие задания.

### **Задание 1**

а) При каких значениях числа  $S$  Петя может выиграть первым ходом? Укажите все такие значения и выигрывающий ход Пети.

б) Укажите одно значение  $S$ , при которых Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

### **Задание 2**

Укажите одно значение  $S$ , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня. Для указанных значений  $S$  опишите выигрышную стратегию Пети.

### Задание 3

Укажите одно значение  $S$ , при котором 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. Для указанных значений  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

**№27.** На вход программы поступает последовательность из  $N$  целых положительных чисел. Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре неважен). Необходимо определить минимальную сумму произвольной пары чисел и количество пар с суммой равной минимальной.

### **Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10000. В качестве результата программа должна вывести два числа: найденную минимальную сумму и количество пар с суммой равной минимальной.

### **Пример входных данных №1:**

**10 1 2 3 1 2 3 1 2 3 1**

### **Выходные данные для приведенного выше примера:**

**2 6**

### **Пример входных данных №2:**

**5                    2                    2                    1                    2                    2**

### **Выходные данные для приведенного выше примера:**

**3 4**

Перед текстом программы кратко опишите алгоритм решения.