



### Владислав Сергеевич Попов

Учитель информатики ГБОУ Цифровая школа, старший преподаватель кафедры «Информационные системы и телекоммуникации» МГТУ им. Н. Э. Баумана.

## Решение заданий 6 и 22 ЕГЭ методом перебора на языке программирования

### Задание № 6

*Постановка задачи:* Определите, при каком наименьшем введённом значении переменной  $s$  программа выведет число 64. Для Вашего удобства программа представлена на четырёх языках программирования (Pascal, Python, Алгоритмический язык, C++). Рассмотрим решение данного типа задач на языках Pascal и Python.

Приведённый в задании текст программы и переборное решение для языка Pascal:

Текст задания на Pascal	Перебор значений $s$ на Pascal
<pre>var s, n: integer; begin   readln(s);   n := 1;   while s &lt; 51 do   begin     s := s + 5;     n := n * 2   end;   writeln(n) end.</pre>	<pre>var s_orig, s, n: integer; begin   for s_orig := -1000 to 1000 do   begin     s := s_orig;     n := 1;     while s &lt; 51 do     begin       s := s + 5;       n := n * 2     end;     if n = 64 then       writeln(s_orig);     end;   end.</pre>

Приведённый в задании текст программы и переборное решение для языка Python:

Текст задания на Python	Перебор значений $s$ на Python
<pre>s = int(input()) n = 1 while s &lt; 51:     s = s + 5     n = n * 2 print(n)</pre>	<pre>for s_orig in range(-1000, 1001):     s = s_orig     n = 1     while s &lt; 51:         s = s + 5         n = n * 2     if n == 64:         print(s_orig)</pre>

Ответ: 21

Обе программы перебора выведут числа 21, 22, 23, 24, 25 – при вводе всех этих чисел программа из задания напечатала бы число 64. В соответствии с заданием экзаменуемому остаётся выбрать и записать в качестве ответа наименьшее из этих чисел – 21.

Для нахождения ответа в приведённом программном коде из задания ввод переменной *s* следует заменить на перебор возможных вводимых значений *s* в цикле `for` с последующей проверкой результата обработки каждого числа. Для организации перебора была создана дополнительная переменная – счётчик цикла *s\_orig*, последовательно принимающая значения в большом диапазоне целых чисел – от -1000 до 1000. В первом выражении тела цикла `for` в переменную *s* присваивается значение переменной *s\_orig*. Значение переменной *s\_orig* в пределах тела цикла будет оставаться неизменным, а значение переменной *s* будет изменяться в цикле `while` как в исходной программе. Поскольку по условию задачи программа должна вывести число 64, вывод переменной *n* был заменён условным оператором, проверяющим условие равенства переменной *n* и числа 64. В случае если переменная *n* имеет значение 64 после окончания работы цикла `while`, выводится значение переменной *s\_orig* – изначальное значение переменной *s* на данной итерации цикла `for`. Изменённая программа выводит все начальные значения переменной *s*, в результате обработки которых в переменной *n* оказалось значение 64.

Пре решении некоторых заданий такого вида необходимо более строго ограничивать диапазон значений счётчика в цикле `for`. Например, для решения следующей задачи начальное значение переменной *s* не может быть отрицательным или нулём (иначе условие цикла всегда будет истинным, поскольку переменная *s* никогда не примет значение, большее или равное 2000).

*Постановка задачи:* Определите, при каком наименьшем введённом значении переменной *s* программа выведет число 60.

<i>Текст задания на Python</i>	<i>Перебор значений s на Python</i>
<pre><b>s = int(input())</b> n = 44 while s &lt; 2000:     s = s * 3     n = n + 4 <b>print(n)</b></pre>	<pre><b>for s_orig in range(1, 1001):</b>     <b>s = s_orig</b>     n = 44     while s &lt; 2000:         s = s * 3         n = n + 4     <b>if n == 60:</b>         <b>print(s_orig)</b></pre>

*Ответ:* 25

Ещё один возможный вариант решения – создание списка или массива подходящих значений *s* с последующим нахождением требуемого ответа в сформированном списке (массиве):

*Перебор значений s на Python*

```

lst = []
for s_orig in range(1, 1001):
    s = s_orig
    n = 44
    while s < 2000:
        s = s * 3
        n = n + 4
    if n == 60:
        lst.append(s_orig)
print("Минимум =", min(lst))
print("Максимум =", max(lst))
print("Количество =", len(lst))

```

**Задание № 22**

Задание № 22 может быть решено схожим образом: вместо ввода одного значения реализуется перебор значений для входной переменной, а вместо вывода результата – проверка значений переменных с выводом начального значения входной переменной.

*Постановка задачи:* 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.

Приведённый в задании текст программы и переборное решение для языка Pascal:

<i>Текст задания на Pascal</i>	<i>Перебор значений x на Pascal</i>
<pre> var x, L, M, Q: integer; begin   readln(x);   Q := 9;   L := 0;   while x &gt;= Q do   begin     L := L + 1;     x := x - Q;   end;   M := x;   if M &lt; L then   begin     M := L;     L := x;   end;   writeln(L);   writeln(M); end. </pre>	<pre> var x_orig, x, L, M, Q: integer; begin   for x_orig := -1000 to 1000 do   begin     x := x_orig;     Q := 9;     L := 0;     while x &gt;= Q do     begin       L := L + 1;       x := x - Q;     end;     M := x;     if M &lt; L then     begin       M := L;       L := x;     end;     if (L = 4) and (M = 5) then       writeln(x_orig);   end; end. </pre>

Приведённый в задании текст программы и переборное решение для языка Python:

<i>Текст задания на Python</i>	<i>Перебор значений <math>x</math> на Python</i>
<pre><b>x</b> = <b>int</b>(<b>input</b>()) Q = 9 L = 0 <b>while</b> x &gt;= Q:     L = L + 1     x = x - Q M = x <b>if</b> M &lt; L:     M = L     L = x <b>print</b>(L) <b>print</b>(M)</pre>	<pre><b>for</b> x_orig <b>in</b> <b>range</b>(-1000, 1001):     <b>x</b> = x_orig     Q = 9     L = 0     <b>while</b> x &gt;= Q:         L = L + 1         x = x - Q     M = x     <b>if</b> M &lt; L:         M = L         L = x     <b>if</b> L == 4 <b>and</b> M == 5:         <b>print</b>(x_orig)</pre>

*Ответ:* 49

Несмотря на возможность простого алгоритмического решения двух рассмотренных заданий, обязательно выполните нахождение ответов вручную: это поможет исключить ошибки в добавленном программном коде и увеличит вероятность верного решения заданий.