

ЕГЭ по информатике 2025 - Задание 5 (Линейный алгоритм)



Привет! В этой статье будут различные примеры решения задач из 5-ого задания ЕГЭ по информатике 2025. Задание 5 решается не сложно, но, как всегда, нужно потренироваться решать подобные задачи, чтобы уверенно себя чувствовать на ЕГЭ по информатике 2025.

Рассмотрим классический пример.

Задача (Классическая)

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R по следующему принципу.

- 1) Строится двоичная запись числа N .
- 2) К этой записи дописываются справа ещё два разряда по следующему правилу:
 - а) Складываются все цифры двоичной записи, 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 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 минимальное число R , которое превышает 42 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 1000): s=format(n, 'b') s=s+str(s.count('1')%2) s=s+str(s.count('1')%2) r=int(s, 2) if r>42: print(r)</pre>	Ответ: 46

Пояснение к задаче.

Программа будет выводить различные числа, но нас интересует самое маленькое. В ответе получается 46. Чтобы остановить поток чисел, можно нажать сочетание **Ctrl + C**.

1. В программе перебираем натуральные числа от 1 до 1000 с помощью цикла **for**. Каждое число подставляем в описанный алгоритм, в надежде получить в результате число r , удовлетворяющие условию задачи.
2. С помощью функции **format** переводим число **n** в двоичный вид. Получаем результат в виде строки **s**.
3. Чтобы найти сумму цифр получившейся двоичной записи, достаточно подсчитать количество единиц в строке **s**. Ведь только единицы в двоичной записи дают в сумму результат. Это можно сделать, применив функцию **.count()** к строке **s**.

4. Добавляем справа к строке **s** остаток от деления суммы цифр на 2. Остаток нужно превратить в строковый тип данных, чтобы "присоединить" его к строке **s** справа.
 5. Повторяем пункт Б, скопировав строку с пунктом А.
 6. Чтобы обратно превратить строку двоичной записи в десятичное число, используем функцию **int()**, указав параметр **2**.
 7. В конце программы пропишем условие. Если **r** больше 42, то будем печатать эти значения.
- Остается выбрать минимальное число **r**. **Ответ:** 46

Задача (Классическая, закрепление)

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

1) Строится двоичная запись числа **N**.

2) К этой записи дописываются справа ещё два разряда по следующему правилу: если **N** чётное, в конце числа справа дописываются два нуля, в противном случае справа дописываются две единицы. Например, двоичная запись 1101 будет преобразована в 110111.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа **N**) является двоичной записью числа - результата работы данного алгоритма.

Укажите минимальное число **N**, для которого результат работы алгоритма будет больше 130. В ответе это число запишите в десятичной системе счисления.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 1000): s=format(n, 'b') if n%2==0: s=s+'00' else: s=s+'11' r=int(s, 2) if r>130: print(n)</pre>	<p>Ответ: Минимальное число n получается 33 Обратите внимание, что здесь уже анализируем число n. Если оно чётное, то к переменной s справа дописываем '00', иначе '11'. Так же в этой задаче мы печатаем в ответе само число n.</p>

Задача (Замена символов)

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

1) Строится двоичная запись числа **N**.

2) Каждый разряд этой записи заменяется двумя разрядами по следующему правилу: если в разряде стоит 0, то вместо него пишется 01; если в разряде стоит 1, то 1 заменяется на 10.

Например, двоичная запись 1010 числа 10 будет преобразована в 10011001.

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

Укажите максимальное нечётное число **R**, меньшее 256, которое может являться результатом работы данного алгоритма. В ответе это число запишите в десятичной системе.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 1000): s=format(n, 'b') s=s.replace('0', '*') s=s.replace('1', '10') s=s.replace('*', '01') r=int(s, 2) if r%2!=0 and r<256: print(r)</pre>	<p>Ответ: получается наибольшее число 169</p> <ol style="list-style-type: none"> Опять с помощью функции <code>format()</code> переводим число <code>n</code> в двоичный вид. С помощью функции <code>replace()</code> заменяем во всей строке <code>s</code> ноль на звёздочку. Таким образом, мы как бы спрятали первоначальные нули. Затем заменяем "1" на "10", и "*" на "01". В результате мы добьёмся нужных замен, о которых говорится в условии задачи.

Похожие задачи встречались в сборнике С. С. Крылова для подготовки к ЕГЭ по информатике.

Задача (Отнимаем остаток)

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

- Из числа `N` вычитается остаток от деления `N` на 4.
- Строится двоичная запись полученного результата.
- К этой записи справа дописываются ещё два дополнительных разряда по следующему правилу:
 - Складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописываются в конец числа (справа). Например, запись 11100 преобразуется в запись 111001.
 - Над этой записью производятся те же действия - справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись является двоичной записью числа `R`.

Укажите **наибольшее** число `N`, для которого результат работы данного алгоритма меньше 47. В ответе число `N` укажите в десятичной системе.

Решение на Python.	Результат работы алгоритма
<pre>for i in range(1, 1000): n = i n = n - n % 4 # Выполняем первый пункт s = format(n, 'b') s = s + str(s.count('1') % 2) # Подпункт а) третьего пункта s = s + str(s.count('1') % 2) # Подпункт б) третьего пункта r = int(s, 2) if r < 47: print(i)</pre>	<p>Ответ: 11</p>

Пояснение к задаче.

- Перебираем числа от 1 до 1000 с помощью цикла `for`.
- В переменную `n` по очереди подставляются числа из нашего диапазона (1 до 1000). Чтобы в ответе была возможность написать первоначальное число, заводим ещё одну переменную `n`, с которой производим основные действия.
- В остальном задача похоже на предыдущие.

Задача (Замена двух левых разрядов)

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

1. Строится двоичная запись числа N.
2. Далее эта запись обрабатывается по следующему правилу:
 - а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;
 - б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа R.

Например, для исходного числа $6_{10} = 110_2$ результатом является число $1000_2 = 8_{10}$, а для исходного числа $4_{10} = 100_2$ результатом является число $1101_2 = 13_{10}$.

Укажите минимальное число N, после обработки которого с помощью этого алгоритма получается число R, большее 40. В ответе запишите это число в десятичной системе счисления.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 1000): s=format(n, 'b') if s.count('1')%2==0: s = s + '0' s = '10' + s[2:] else: s = s + '1' s = '11' + s[2:] r=int(s, 2) if r>40: print(n)</pre>	<p>Ответ: 16</p> <p>1. Здесь пишем программу, как в предыдущих примерах. Но, действительно, встречается и новый приём.</p> <p>2. Нужно изменить левые символы нашей строки s. Это можно сделать с помощью такой конструкции s[2:]. Таким образом, мы берём всю строку, кроме двух первых символов.</p> <p>3. Например, s='football', то s[2:] будет обозначать 'otball'.</p>

Пояснение к задаче.

1. Перебираем числа от 1 до 1000 с помощью цикла **for**.
2. В этом диапазоне надеемся найти наш ответ.
3. С помощью команды **format()** превращаем число в строку уже в двоичной системе.
4. Сумма цифр в строке зависит только от количества единиц.
5. Нули ничего не дают в сумму. Поэтому применяем функцию **.count**.
6. Дальше всё делаем, как написано в условии задачи. Команда **int(s, 2)** переводит строку из двоичной системы в десятичное число. **Ответ: 16**

Задача (Работаем с символами строки)

Автомат обрабатывает натуральное число N > 1 по следующему алгоритму:

- 1) Строится двоичная запись числа N.
- 2) В конец записи (справа) дописывается вторая справа цифра двоичной записи.
- 3) В конец записи (справа) дописывается вторая слева цифра двоичной записи.
- 4) Результат переводится в десятичную систему.

Пример. Дано число N = 11. Алгоритм работает следующим образом.

- 1) Двоичная запись числа N: $11 = 1011_2$

2) Вторая справа цифра 1, новая запись 101112.

3) Вторая слева цифра 0, новая запись 1011102.

4) Десятичное значение полученного числа 46.

При каком наименьшем числе N в результате работы алгоритма получится R > 170? В ответе запишите это число в десятичной системе счисления.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(2, 1000): s=format(n, 'b') s=s[-2] s=s+s[1] r=int(s, 2) if r>170: print(n)</pre>	<p>Ответ: 43. Получается наименьшее число 43</p> <ol style="list-style-type: none">К последнему символу можем обратиться как s[-1].К предпоследнему как s[-2].Но счёт слева начинается с нулевого индекса.Первый символ - это s[0], второй символ - это s[1] и т.д.

Обратите внимание, что перебирать числа **n** в этой задаче начинаем с 2. Это делается, чтобы в двоичной системе число имело минимум два разряда. Если начнём с 1, то программа не сможет обратиться к предпоследнему символу и выдаст ошибку! **Ответ: 43**

Задача (Восьмибитное число)

Автомат обрабатывает натуральное число N ($1 \leq N \leq 255$) по следующему алгоритму:

1) Строится восьмибитная двоичная запись числа N.

2) Удаляется последняя цифра двоичной записи.

3) Запись «переворачивается», то есть читается справа налево.

4) Полученное число переводится в десятичную запись и выводится на экран.

Каково наибольшее число, меньшее 100, которое после обработки автоматом не изменится?

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 256): s=format(n, 'b') # делаем 8-ое число while(len(s)<8): s='0'+s s=s[:-1] # удаляется последняя цифра s=s[::-1] # число переворачивается r=int(s, 2) if n<100 and r==n: print(n)</pre>	<p>Ответ: 90</p>

Пояснение к задаче.

- Восьмибитное число имеет длину 8 символов. После того, как перевели число **n** в двоичный вид, с помощью цикла **while** добываем нули слева к строке **s**, пока длина этой строки меньше 8.
- Удалить последнюю цифру можно с помощью конструкции **s[:-1]**. Здесь мы оставляем все цифры, начиная с первой до последней (не включительно).
- Перевернуть строку можно с помощью конструкции **s[::-1]**.
- Далее решаем как обычно.
- Если при перевороте строки получаются незначащие нули, то в этом ничего страшного нет. Функция **int()** их просто отбрасывает.
- Число не изменится, если входное число **n** равно выходному числу **r**.

Задача (Найти N при наименьшем R)

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

1. Строится двоичная запись числа N.

2. Далее эта запись обрабатывается по следующему правилу:

а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 1;

б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11;

Полученная таким образом запись является двоичной записью искомого числа R. Укажите число N, после обработки которого с помощью этого алгоритма получается наименьшее значение R, большее 49. В ответе запишите это число в десятичной системе.

Например, для исходного числа $6_{10} = 110_2$ результатом является число $100_2 = 4_{10}$, а для исходного числа $4_{10} = 100_2$ результатом является число $1101_2 = 13_{10}$.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1, 1000): s=format(n, 'b') if s.count('1')%2==0: s = s + '0' s = '1' + s[2:] else: s = s + '1' s = '11' + s[2:] r=int(s, 2) if r>49: print(r, n)</pre>	<p>99 49 101 50 105 52 111 55 113 56 <u>50 57</u> ← 52 58 119 59 56 60 123 61 125 62 62 63 193 64</p>

Пояснение к задаче.

- Хитрость задачки заключается в том, что числа r возрастают неравномерно.
- Нам необходимо глазами найти наименьшее число r (первое число). Это число 50, а n для него равно 57.
- При желании программу можно переписать следующим образом:

Решение на Python.	Результат работы алгоритма
<pre>r_min=10**9 n_r_min = 0 for n in range(1, 1000): s=format(n, 'b') if s.count('1')%2==0: s = s + '0' s = '1' + s[2:] else: s = s + '1' s = '11' + s[2:] r=int(s, 2) if r > 49: if r < r_min: r_min=r</pre>	<p>Ответ: 57 Здесь ищется минимальное число r автоматически и для него запоминается значение n, которое пойдет в ответ. Этот приём условно можно назвать "царь горы".</p>

<pre>n_r_min=n print(n_r_min)</pre>	
-------------------------------------	--

Задача (Другая система счисления)

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

1. Строится троичная запись числа N .

2. Далее эта запись обрабатывается по следующему правилу:

a) если число N делится на 3, то к этой записи дописываются две последние троичные цифры;

б) если число N на 3 не делится, то остаток от деления умножается на 5, переводится в троичную запись и дописывается в конец числа.

Полученная таким образом запись является троичной записью искомого числа R .

3. Результат переводится в десятичную систему и выводится на экран. Укажите минимальное число R , большее 133, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной системе счисления.

Например, для исходного числа $11 = 102_3$ результатом является число $102101_3 = 307$, а для исходного числа $12 = 110_3$ это число $11010_3 = 111$.

Решение на Python.	Результат работы алгоритма
<pre>def F(n): s="" while n>0: s=s+str(n%3) n=n//3 return s[::-1] mn_r=10**9 for n in range(1, 1000): s=F(n) if n%3==0: s = s + s[-2] + s[-1] if n%3!=0: s = s + F((n%3)*5) r=int(s, 3) if r>133: mn_r = min(mn_r, r) print(mn_r)</pre>	<p>Ответ: 141</p> <p>Функция получает число, переводит его в троичную систему, возвращает результат в виде строки.</p> <p>Здесь вместо функции <code>format</code> нам придётся создать собственную функцию <code>F</code>. Эта функция должна перевести число n в троичную систему и вернуть результат в виде строки.</p>

Пояснение к задаче.

1. Рассмотрим эту функцию более подробно. Аналогично переводили из десятичной системы в любую другую в 9 классе делением уголком.
2. Заводим строку s . Пока в переменной n что-то есть, продолжаем переводить в троичную систему.
3. Остаток от деления на 3 - это и есть очередная цифра нашего числа в троичной системе, начиная с конца. Но нужно ещё так же делить целочисленным образом число n на 3. В 9 классе, когда мы переводили уголком, так же делили наше число, и оно становилось всё меньше и меньше.

4. После окончания работы цикла `while`, в строке `s` будет нужное нам число в троичной системе, но записанное слева направо. Нужно перевернуть эту строку `s[::-1]`, перед тем, как функция вернёт её.
5. Подобная функция хорошо работает, когда основание системы, куда мы переводим, меньше 10. В противном случае у нас может получится, к примеру, остаток 11, и трудно будет отличить, где число 11, а где просто две единицы. В этом случае можно попробовать дописать в функцию дополнительные условия.
6. Далее действуем, как в предыдущих задачах. Так же используем приём "царь горы", т.к. числа `r` получаются неравномерными.

Задача (Добавляем символ в середину)

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

1. Строится четверичная запись числа N .

2. Далее эта запись обрабатывается по следующему правилу:

a) если количество значащих цифр в четверичной записи числа чётное, то к этой записи в середину дописывается 0;

b) если количество значащих цифр в четверичной записи числа нечётное, то к этой записи справа дописывается 1;

Полученное число переводится в десятичную запись и выводится на экран.

Какое наименьшее число, превышающее 40, может получиться в результате работы автомата?

Решение на Python.	Результат работы алгоритма
<pre>def F(n): s="" while n>0: s=s+str(n%4) n=n//4 return s[::-1] mn_r=10**9 for n in range(1, 1000): s=F(n) if len(s)%2==0: s = s[:len(s)//2] + '0' + s[len(s)//2:] else: s = s + '1' r=int(s, 4) if r>40: mn_r = min(mn_r, r) print(mn_r)</pre>	<p>Ответ: 48</p> <ol style="list-style-type: none"> 1. Проверить количество символов в строке <code>s</code> можно с помощью функции <code>len()</code>. 2. Так <code>s[:len(s)//2]</code> можно получить первую половину строки, так <code>s[len(s)//2:]</code> - вторую половину. 3. Далее решаем, как в прошлых задачах. 4. Ответ: 48

Боковой вариант 5-ого задания из ЕГЭ по информатике.

Задача (Лучше знать)

Автомат получает на вход четырёхзначное число. По этому числу строится новое число по следующим правилам:

1. Перемножаются первая и вторая, а также третья и четвёртая цифры исходного числа.

2. Полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 2465. Суммы: $2 * 4 = 8$; $6 * 5 = 30$. Результат: 308. Укажите наибольшее число, в результате обработки которого автомат выдаст число 124.

Решение:

В подобных задачах из ЕГЭ по информатике нумерация происходит начиная со старшего разряда.

1	2	3	4
---	---	---	---

Первое правило можно представить следующим образом:

 - первое число

 - второе число

Второе правило заключается в том, что мы "соединяем" два числа, полученных в первом пункте, причём, сначала идёт **большее** число, а затем **меньшее**.

Проанализируем число 124.



Чтобы четырёхзначное число было наибольшим, выгодно, чтобы в старшем разряде стояла 9. Но, не у числа 12, не у числа 4, нет такого делителя. Какой наибольший делитель мы можем получить? Это число 6. Число 6 является делителем **12-ти**. Значит, первая цифра будет 6, а вторая цифра будет 2 ($6 * 2 = 12$).

Рассмотрим второе число 4. Третий разряд тоже желательно сделать побольше. Значит, в четвёртый разряд поставим 4, а в младший разряд 1 ($4 * 1 = 4$). Ответ получается **6241**.

Решение на Python.	Результат работы алгоритма
<pre>for n in range(1000, 10000): s = str(n) x = int(s[0])*int(s[1]) y = int(s[2])*int(s[3]) if x>y: r = str(x) + str(y) else: r = str(y) + str(x) if r == '124': print(n)</pre>	<p>Ответ: 6241.</p> <ol style="list-style-type: none">В начале переводим число n в строку. Разбиваем число по цифрам и получаем нужных два числа.Соединяем эти два числа в порядке убывания. Удобно соединить, превратив опять эти два числа в строчный тип данных.Проверяем результат под ответ. Важно не забыть, что переменная r в этом решении имеет строчный тип данных. Ответ: 6241

Материал взят с сайта: https://code-enjoy.ru/ege_po_informatike_2025_zadanie_5_lineniy_algoritm/