



Попов Владислав Сергеевич

Старший преподаватель кафедры

«Информационные системы и телекоммуникации»

МГТУ им. Н.Э. Баумана

Решаем новую задачу ЕГЭ № 14 в Python и электронной таблицей

Функция `int` и полиномиальное представление чисел

В демонстрационном варианте ЕГЭ по информатике 2023 года была представлена новая задача № 14 на арифметические операции в системах счисления:

«Операнды арифметического выражения записаны в системе счисления с основанием 15.

$$123x5_{15} + 1x233_{15}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 15-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 14. Для найденного значения x вычислите частное от деления значения арифметического выражения на 14 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.»

Решим эту задачу несколькими способами на компьютере.

Способ 1. Перевод чисел с помощью функции `int` в десятичную систему счисления

```
digits = '0123456789ABCDE'  
for x in digits:  
    number = int('123' + x + '5', 15) + int('1' + x + '233', 15)  
    if number % 14 == 0:  
        print(number // 14)
```

Вывод программы:

8767

10462

Поскольку 15-ричные цифры x перебирались в порядке возрастания, в ответе на задание для наименьшего подходящего x следует указать первое частное от деления суммы на 14, ответом будет число 8767.

В первой строке программы в переменную `digits` присваивается строка, состоящая из цифр 15-ричной системы счисления – от 0 до E: `digits = '0123456789ABCDE'`. В цикле `for` происходит перебор символов в строке `digits`: в переменную `x` присвоится сначала символ '0', затем символ '1', ..., последний присвоенный символ будет символом 'E'. Для каждого из символов на отдельной итерации цикла `for` в строке программы `number = int('123' + x + '5', 15) + int('1' + x + '233', 15)` вычисляется десятичное значение выражения $123x5_{15} + 1x233_{15}$. Обратите внимание на функцию `int`: первый аргумент является строкой и получается в результате конкатенации (слияния) строк, где переменная `x` тоже является строкой ('123' + `x` + '5'), а второй аргумент функции `int(15)` является основанием системы счисления числа, представленного первым аргументом, функция `int` возвращает целое число в десятичной системе счисления. Если полученное десятичное значение `number` делится на 14 без остатка (`if number % 14 == 0:`), то программа выводит значение частного от целочисленного деления значения переменной `number` на 14 (`number // 14`).

Для дополнительной проверки возможен вывод результата не целочисленного, а обычного деления значения переменной `number` на 14 (`number / 14`), в этом случае вывод программы будет состоять из двух вещественных чисел: 8767.0 и 10462.0, в ответе на данное задание нужно записать первое число без нулевой дробной части. Также, хотя строковое значение записывается гораздо короче, возможно задание переменной `digits` в виде списка символов-цифр: `digits = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E']`.

Возможна модификация приведённого программного кода для решения задачи без применения условного оператора:

```
digits = '0123456789ABCDE'
for x in digits:
    number = int('123' + x + '5', 15) + int('1' + x + '233', 15)
    print(x, number / 14)
```

Вывод программы:

```
0 7798.428571428572
1 8040.571428571428
2 8282.714285714286
3 8524.857142857143
4 8767.0
5 9009.142857142857
6 9251.285714285714
7 9493.42857142857
8 9735.57142857143
9 9977.714285714286
A 10219.857142857143
B 10462.0
C 10704.142857142857
D 10946.285714285714
E 11188.42857142857
```

В приведённой программе осуществляется вывод каждой цифры x в порядке возрастания и соответствующего частного от деления десятичного числа на 14. В ответ нужно записать целую часть от первого частного с нулевой дробной частью (8767). Обратите внимание, что для решения задачи был осуществлён вывод вещественного частного от деления значения переменной `number` на 14 (`number / 14`).

Ограничение функции `int`

Функция `int`, используемая в приведённых программах для перевода чисел из 15-ричной системы счисления в десятичную, имеет существенное ограничение: в качестве второго параметра – основания системы счисления – она принимает значение в диапазоне от 2 до 36, значение параметра по умолчанию равняется 10. Верхняя граница диапазона второго аргумента, равная 36, получается как сумма 10 десятичных цифр (0-9) и 26 букв английского алфавита (A-Z), для систем счисления с большим основанием нет договорённости о цифрах, идущих после цифры Z, что обуславливает рассматриваемое ограничение.

Так, в случае вызова `int('12345', 36)` функция вернёт десятичное значение $1776965_{10} = 12345_{36} = 1 \cdot 36^4 + 2 \cdot 36^3 + 3 \cdot 36^2 + 4 \cdot 36^1 + 5 \cdot 36^0$. В случае вызова функции `int('12345', 37)` будет сгенерировано исключение: `ValueError: int() base must be >= 2 and <= 36, or 0`. Поэтому при основании системы счисления в задании, большем чем 36, использование первого способа решения задачи невозможно.

Способ 2. Перебор десятичного представления 15-ричных цифр и полиномиальная запись чисел

Представим числа $123x5_{15}$ и $1x233_{15}$ в виде полиномов:

$$123x5_{15} = 1 \cdot 15^4 + 2 \cdot 15^3 + 3 \cdot 15^2 + x \cdot 15^1 + 5 \cdot 15^0$$

$$1x233_{15} = 1 \cdot 15^4 + x \cdot 15^3 + 2 \cdot 15^2 + 3 \cdot 15^1 + 3 \cdot 15^0$$

При использовании полиномиальной записи чисел можно перебрать все возможные цифры x в виде их десятичного представления, получить десятичное значение выражения $123x5_{15} + 1x233_{15}$, после чего проверить делимость каждой полученной десятичной суммы значений на 14 и вывести частное в случае делимости:

```
for x in range(15):
    number = (1 * 15**4 + 2 * 15**3 + 3 * 15**2 + x * 15 + 5) +
    (1 * 15**4 + x * 15**3 + 2 * 15**2 + 3 * 15 + 3)
    if number % 14 == 0:
        print(number // 14)
```

Цикл `for x in range(15)` позволяет перебрать целые числа в диапазоне от 0 до 14 включительно, что соответствует десятичному значению 15-ричных цифр от 0 до E.

Хорошей практикой программирования является использование переменных и именованных констант вместо литеральных констант или просто лите-

ралов – значений, которые вставляются непосредственно в программный код и иронично называются magic constants. Запись основания системы счисления в отдельную переменную уменьшает вероятность ошибки при вычислении значения полинома:

```
base = 15
for x in range(base):
    number = (1 * base**4 + 2 * base**3 + 3 * base**2 + x * base + 5) + (1 * base**4 + x * base**3 + 2 * base**2 + 3 * base + 3)
    if number % 14 == 0:
        print(number // 14)
```

Для решения задачи в одну строку на языке Python можно воспользоваться такой синтаксической особенностью языка как list comprehension:

```
base = 15; [print(x, ((1 * base**4 + 2 * base**3 + 3 * base**2 + x * base + 5) + (1 * base**4 + x * base**3 + 2 * base**2 + 3 * base + 3)) / 14) for x in range(base)]
```

Данный программный код выведет значения x и значения соответствующих выражений, разделенные на 14.

Аналогичный метод можно применить для решения задачи в электронных таблицах, записав в первом столбце все возможные десятичные представления цифр (для 15-ричной системы счисления – числа от 0 до 14), во втором столбце – выражение (например, $(1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A2 * 15 + 5) + (1 * 15^4 + A2 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$), а в третьем столбце – искомое частное (например, $=B2 / 14$):

	A	B	C
1	Цифра	Значение выражения	Частное (значение / 14)
2	0	109178	7798,428571
3	1	112568	8040,571429
4	2	115958	8282,714286
5	3	119348	8524,857143
6	4	122738	8767
7	5	126128	9009,142857
8	6	129518	9251,285714
9	7	132908	9493,428571
10	8	136298	9735,571429
11	9	139688	9977,714286
12	10	143078	10219,85714
13	11	146468	10462
14	12	149858	10704,14286
15	13	153248	10946,28571
16	14	156638	11188,42857

Обратите внимание, что в формулах второго столбца для расчёта значений выражения $123x_{15} + 1x_{233}_{15}$ вместо переменной x в каждой строке указана ссылка на соответствующее число из первого столбца. Если в Python операция возведения в степень обозначается двумя астерисками (**), то в электронных таблицах – символом карет (^). Нужно заметить, что символ ^ во многих языках программирования обозначает операцию побитового исключающего или, в том числе и в Python, поэтому эту операцию нельзя использовать как замену операции ** в Python. Также заметим, что операция возведения в степень является более приоритетной, чем операция умножения, поэтому операцию возведения в степень при решении данной задачи не нужно заключать в скобки.

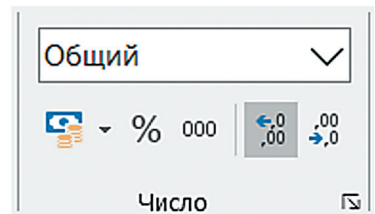
Для вычисления значений выражения и частного были использованы следующие формулы:

	A	B	C
1	Цифра	Значение выражения	Частное (значение / 14)
2	0	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A2 * 15 + 5) + (1 * 15^4 + A2 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=82 / 14
3	1	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A3 * 15 + 5) + (1 * 15^4 + A3 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=83 / 14
4	2	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A4 * 15 + 5) + (1 * 15^4 + A4 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=84 / 14
5	3	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A5 * 15 + 5) + (1 * 15^4 + A5 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=85 / 14
6	4	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A6 * 15 + 5) + (1 * 15^4 + A6 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=86 / 14
7	5	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A7 * 15 + 5) + (1 * 15^4 + A7 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=87 / 14
8	6	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A8 * 15 + 5) + (1 * 15^4 + A8 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=88 / 14
9	7	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A9 * 15 + 5) + (1 * 15^4 + A9 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=89 / 14
10	8	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A10 * 15 + 5) + (1 * 15^4 + A10 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=90 / 14
11	9	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A11 * 15 + 5) + (1 * 15^4 + A11 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=91 / 14
12	10	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A12 * 15 + 5) + (1 * 15^4 + A12 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=92 / 14
13	11	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A13 * 15 + 5) + (1 * 15^4 + A13 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=93 / 14
14	12	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A14 * 15 + 5) + (1 * 15^4 + A14 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=94 / 14
15	13	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A15 * 15 + 5) + (1 * 15^4 + A15 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=95 / 14
16	14	$= (1 * 15^4 + 2 * 15^3 + 3 * 15^2 + A16 * 15 + 5) + (1 * 15^4 + A16 * 15^3 + 2 * 15^2 + 3 * 15 + 3)$	=96 / 14

Формулы из ячеек B2 и C2 скопированы в следующие строки соответствующих столбцов.

Для отображения большего количества знаков после запятой следует выделить соответствующие ячейки и нажать кнопку «Увеличить разрядность» группы «Число» вкладки «Главное».

Кроме того, для решения задачи можно использовать функцию вычисления остатка от целочисленного деления ОСТАТ(делимое; делитель).



Интерактивный режим Python

Возможно решение представленной задачи в интерактивном режиме Python в IDLE при записи минимального количества программного кода:

```
>>> int('12305', 15) + int('10233', 15)
109178
>>> 109178 / 14
7798.428571428572
>>> int('12315', 15) + int('11233', 15)
112568
>>> 112568 / 14
8040.571428571428
```

```

>>> int('12325', 15) + int('12233', 15)
115958
>>> 115958 / 14
8282.714285714286
>>> int('12335', 15) + int('13233', 15)
119348
>>> 119348 / 14
8524.857142857143
>>> int('12345', 15) + int('14233', 15)
122738
>>> 122738 / 14
8767.0
>>> |

```

При решении задачи в интерактивном режиме происходит вычисление значения выражения $123x5_{15} + 1x233_{15}$ для каждого x до тех пор, пока частное при делении значения выражения на 14 не станет целым числом, которое нужно записать в ответ без дробной части (8767).

Поскольку при написании программы можно получить неверный ответ, допустив и не заметив опечатку даже в одном символе, рекомендуется проверить собственное решение в электронных таблицах рассмотренным способом и, по возможности, провести проверку самостоятельным решением без использования программирования или электронных таблиц.

Дополнительные задания для самостоятельного решения

1. Операнды арифметического выражения записаны в системе счисления с основанием 11.

$$1A2x3_{11} + x987A_{11}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 11-ричной системы счисления. Определите значение x , при котором значение данного арифметического выражения кратно 35. Для найденного значения x вычислите частное от деления значения арифметического выражения на 35 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

Ответ: 1178

2. Операнды арифметического выражения записаны в системе счисления с основанием 40.

$$7AZx5_{40} + xABCD_{40} - 1234x_{40}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 40-ричной системы счисления. Определите наибольшее значение x , при котором значение данного арифметического выражения кратно 15. Для найденного значения x вычислите частное от деления значения арифметического выражения на 15 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

Ответ: 6737495

3*. Операнды арифметического выражения записаны в системе счисления с основанием 59.

$$xIBMx_{59} - HP12x_{59}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 59-ричной системы счисления. Определите наименьшее и наибольшее значение x , при котором значение данного арифметического выражения кратно 10. Для найденных значений x вычислите модуль разности соответствующих выражений и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

Ответ: 605868050

4*. Операнды арифметического выражения записаны в системе счисления с основаниями $(x + 1)$ и 64.

$$9xxxx_{(x+1)} + xxxx8_{64}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 64-ричной системы счисления. Определите количество цифр x , для которых данная сумма будет нечётной.

Ответ: 55

5*. Операнды арифметического выражения записаны в системе счисления с основанием 64.

$$YxxZx_{64} + AxxVx_{64} + x^x$$

В записи чисел переменной x обозначена неизвестная положительная цифра из алфавита 64-ричной системы счисления. Определите количество цифр x , для которых данная сумма будет чётной.

Ответ: 31

МУДРЫЕ МЫСЛИ

«Отладка кода вдвое сложнее, чем его написание. Так что если вы пишете код настолько умно, насколько можете, то вы по определению недостаточно сообразительны, чтобы его отлаживать».

Brian W. Kernighan

«Большинство программ на сегодняшний день подобны египетским пирамидам из миллиона кирпичиков друг на друге и без конструктивной целостности — они просто построены грубой силой и тысячами рабов».

Alan Kay

«Если вы дадите человеку программу, то займете его на один день. Если вы научите человека программировать, то займете его на всю жизнь».

Waseem Latif

«Иногда лучшие программы создаются на бумажке. Запрограммировать их — второстепенная вещь».

Max Kanat-Alexander