

**Решение заданий №7, №11,
№14 ЕГЭ по информатике
на языке
программирования Python**

Теория: функции **int**, **round**, **ceil**, **log2**

- **int** – округление в сторону нуля, отбрасывание дробной части;
- **round** – округление до ближайшего целого;
- **ceil** – (из модуля `math`) округляет вещественное число **x** вверх к ближайшему целому, не меньше **x**);
- **log2** – (из модуля `math`) вычисляет двоичный логарифм числа.

Вызываются функции с помощью следующих конструкций:

from math import ceil и **from math import Log2** ИЛИ **from math import ceil, Log2**

Теория: особенность функции **int**

➤ Функция **int** может принимать второй (необязательный) аргумент – основание системы счисления (СС) (от 2 до 36). Результат – число в 10-й СС, например:

➤ `n1=int('377',8)`

`print(n1)` # 255

➤ `n2=int('377',9)`

`print(n2)` # 313

➤ `n3=int('377',18)`

`print(n3)` # 1105

Получить в интерактивной консоли все 3 результата нельзя, только 1.

Практика Задача (Тип 11 № 13462 с сайта РешуЕГЭ)

- ▶ При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из **40 символов** и содержащий цифры и прописные символы латинского алфавита (всего **36 возможных символов**). В базе данных для хранения сведений о каждом пользователе отведено одинаковое минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт, одно и то же для всех пользователей. Для хранения сведений о **20** пользователях потребовалось **800 байт**. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Практика: Решение в интерактивной оболочке Python

- ▶ Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
- ▶ Type "help", "copyright", "credits" or "license()" for more information.
- ▶ 36 {мощность алфавита $M=36$ }
- ▶ 36
- ▶ from math import log2,ceil {вызываем функции}
- ▶ ceil(log2(_)) {по формуле $i \geq \log_2(M)$, (**_**) – это предыдущий результат }
- ▶ 6
- ▶ ceil(40*_/8) { находим информационный объем 40 символов (один пароль) и переводим в байты }
- ▶ 30
- ▶ 800/20-_
{из формулы $20*(x+30)=800$, где x – это неизвестный инф.объем доп.сведений }
- ▶ **10.0**

Ещё одна задача Тип 11 № 18082

- ▶ Для регистрации на сайте некоторой страны пользователю необходимо придумать пароль длиной ровно **10** символов. В пароле можно использовать только прописные буквы английского алфавита, то есть **26** символов. Информация о пользователе хранится с помощью минимально возможного целого количества байт. Каждый символ в пароле кодируется одинаковым и минимально возможным количеством бит. Для хранения дополнительной информации на одного пользователя отводится **15 байт. Определите объем памяти в байтах, необходимый для хранения информации о 50 пользователях.**
- ▶ Дано:
- ▶ На 1 пароль – 10 симв.; $M=26$; для доп.свед. 1 пользователя – 15 байт.

Решение в интерактивной консоли Python

- ▶ Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
- ▶ Type "help", "copyright", "credits" or "license()" for more information.
- ▶ 26
- ▶ 26
- ▶ from math import log2,ceil
- ▶ ceil(log2(_))
- ▶ 5
- ▶ ceil(10*_ /8)
- ▶ 7
- ▶ _+15
- ▶ 22
- ▶ _*50
- ▶ **1100**

Задача Тип 7 (передача графического изображения и аудио)

- ▶ Николай отправил другу фотографию в цветовой модели RGB (в модели RGB есть **три основных цвета**, красный, синий, зеленый, у каждого из них по **256** оттенков, комбинации оттенков позволяют получить любой цвет) размером **2048 × 1024** пикселей и **двухканальный** аудиофайл с **32-битным** разрешением и частотой дискретизации **64** кГц. Фотография пришла другу Николаю через **10,24** секунды, а аудиофайл через **100** секунд.
- ▶ **Сколько секунд длиться аудиозапись**, если оба файла были переданы по одному каналу связи? В ответе укажите только целое число. Единицы измерения писать не нужно.

Теоретическое решение (начало)

Для хранения растрового изображения нужно выделить в памяти


$I = K \cdot i$ бит, где K – количество пикселей и i – количество бит, отводимое на 1 пиксель.

- ▶ Глубина кодирования – это количество бит, которые выделяются на хранение цвета одного пикселя. При глубине кодирования i бит на пиксель, код каждого пикселя выбирается из $M=2^i$ возможных вариантов, где N – количество цветов в палитре.
- ▶ Так как используется цветовая модель RGB (256 · 256 · 256 цветов), $i = \log_2(256 \cdot 256 \cdot 256) = 8 + 8 + 8 = 24$ бит; $K = 2048 \cdot 1024 = 2^{21}$ пикселей.
- ▶ Подставим известные значения в формулу: $I = K \cdot i$ и найдем инф.объем изображения:
- ▶ $I = 2^{21} \cdot 24 = 2^{24} \cdot 3$ бит / $2^{13} = 2^{11} \cdot 3$ Кбайт
- ▶ Так как фотография пришла через $10,24 = 256/25 = 2^8 / 5^2$ секунд, скорость передачи данных в данном канале составляет $V_n = I/t = 2^{11} \cdot 3 / 2^8 / 5^2 = 2^3 \cdot 3 \cdot 5^2$ Кбайт/с
- ▶ Откуда размер аудио файла составляет:
- ▶ $100 \cdot 2^3 \cdot 3 \cdot 5^2 = 2^5 \cdot 3 \cdot 5^4$ Кбайт.

Теоретическое решение(окончание)

- Для хранения информации о звуке длительностью t секунд, закодированном с частотой дискретизации f Гц и глубиной кодирования B бит с k каналами записи требуется $I = t \cdot f \cdot B \cdot k$ бит памяти.
- f (Гц) - частота дискретизации определяет количество отсчетов, запоминаемых за 1 секунду.
- B (бит) - глубина кодирования - это количество бит, которые выделяются на один отсчет.
- Итак, выпишем что нам **дано**:
- $t = x$ с; $k = 2$ канала; $B = 32$ бит; $f = 64$ кГц = $64 \cdot 1000$ Гц
- $I = 2^5 \cdot 3 \cdot 5^4 \cdot 2^{13} = 2^{18} \cdot 3 \cdot 5^4$ бит. (переводим Кбайты в биты)
- Подставим всё, что дано, в формулу $I = t \cdot f \cdot B \cdot k$:
- $2^{18} \cdot 3 \cdot 5^4 = t \cdot 64 \cdot 1000 \cdot 32 \cdot 2 = t \cdot 2^{15} \cdot 5^3 \Rightarrow t = 2^3 \cdot 3 \cdot 5 = 120$ с.
- Ответ: **120**

Решение на Python(в консоли)



```
➤ 256*256*256
➤ 16777216
➤ from math import log2
➤ log2(_)
➤ 24.0
➤ 2048*1024*(_)/2**13
➤ 6144.0
➤ (_)/10.24
➤ 600.0
➤ (_)*100
➤ 60000.0
➤ (_)*2**13
➤ 491520000.0
➤ (_)/(64*1000*32*2)
➤ 120.0
```

Теория: функции перевода в различные СС

Встроенные функции **bin**, **oct** и **hex** нужны для перевода числа в 2-ю, 8-ю и 16-ю СС соответственно:

```
c2=bin(270)
```

```
print(c2)      # 0b100001110
```

```
c8=oct(270)
```

```
print(c8)      # 0o416
```

```
c16=hex(270)
```

```
print(c16)     # 0x10e
```

► Чтобы избавиться от ненужных символов в начале строки, кроме срезов ([2:]), используют форматированные строки (или **f-строки**, англ. formatted string literals), в которых для получения двоичных, восьмеричных и шестнадцатеричных кодов чисел применяются соответственно форматы b, o и x (или X):

```
► s2 = f"{255:b}" # "11111111"
```

```
► s8 = f"{255:o}" # "377"
```

```
► s16 = f"{255:x}" # "ff"
```

```
► S16 = f"{255:X}" # "FF"
```

Особенности f-строки

- ▶ В f-строке можно задать количество позиций для вывода числа, а свободные позиции при необходимости заполнить цифрой 0:
- ▶ `s2 = f"{255:010b}"` # "0011111111"
- ▶ `s8 = f"{255:05o}"` # "00377"
- ▶ `s16 = f"{255:06x}"` # "0000ff"
- ▶ `S16 = f"{255:04X}"` # "00FF"

Задача типа 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 СС с основанием 16. Сколько **значащих нулей** содержится в этой записи?

Решение:

```
s=3*4**38+2*4**23 +4**20+3*4**5+2*4**4+1
```

```
h=f'{s:X}' #или h=hex(s)[2:] # форматная строка или функция hex
```

```
print(h.count('0')) # метод count позволяет подсчитать кол-во определенных символов в строке
```

Другое решение через цикл:

1. Перебор индексов всех символов строки.

```
s=3*4**38+2*4**23 +4**20+3*4**5+2*4**4+1
```

```
h=hex(s)[2:]
```

```
count=0
```

```
for i in range(len(h)):
```

```
    if h[i]=='0':
```

```
        count+=1
```

```
print(count)
```

➤ 2. Перебор самих символов

➤ $s=3*4^{38}+2*4^{23}+4^{20}+3*4^5+2*4^4+1$

➤ `h=hex(s)[2:]`

➤ `count=0`

➤ `for s in h:`

➤ `if s=='0':`

➤ `count+=1`

➤ `print(count)`

Ответ: 15

Другие типы задач 14 будем решать с помощью цикла.

Алгоритм деления числа на 16 и составление результата из остатков от деления

```
count = 0
while n > 0:
    if n % 16 == 0:
        count += 1
    n // 16
print(count)
```




Спасибо за внимание!

«Единственный способ выучить новый язык программирования — написать на нем программу».