

Задача 1 (Классическая)

Женя составляет 5-буквенные слова, в которых встречаются только буквы А, Б, В, Г, причём буква А появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Женя?

Решение:

Напишем программу на языке **Python**.

```
k=0
for x1 in 'АБВГ':
    for x2 in 'АБВГ':
        for x3 in 'АБВГ':
            for x4 in 'АБВГ':
                for x5 in 'АБВГ':
                    s=x1+x2+x3+x4+x5
                    if s.count('А')==1:
                        k=k+1
print(k)
```

Т.к. слова состоят из **5-ти** символов, то мы формируем **пять** вложенных циклов! В каждом цикле перебираем все буквы, которые нам дали.

Внутри циклов мы составляем само слово в переменной **s**. Таким образом, в переменной **s** "прокрутятся" все возможные комбинации.

Но мы подсчитываем не все комбинации, а только те, где всего одна буква А.

Важно не перепутать русские и английские буквы.

Ответ: 405

Задача 2 (Каждую букву можно использовать один раз)

Артур составляет 5-буквенные коды из букв Е, С, А, У, Л. Каждую букву нужно использовать ровно один раз, при этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?

Решение:

Запрограммируем решение этой задачи на Питоне.

```
k=0
for x1 in 'ЕСАУЛ':
    for x2 in 'ЕСАУЛ':
        for x3 in 'ЕСАУЛ':
            for x4 in 'ЕСАУЛ':
                for x5 in 'ЕСАУЛ':
                    s=x1+x2+x3+x4+x5
                    if s.count(x1)==1 and s.count(x2)==1 and s.count(x3)==1 and
s.count(x4)==1 and s.count(x5)==1:
                        if s.count('ЕА')==0 and s.count('АЕ')==0 and s.count('ЕУ')==0 and
s.count('УЕ')==0 and s.count('АУ')==0 and s.count('УА')==0:
                            k=k+1
print(k)
```

В первом условии учли, что каждая буква встречается в слове только один раз. Второе условие говорит о том, что две гласные не стоят рядом (перебрали все возможные сочетания гласных).

Ответ: 12

Задача 3 (Буквы составляют перестановкой)

Петя составляет шестибуквенные слова перестановкой букв слова КАБАЛА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?

Решение:

```
k=0
for x1 in 'КБЛА':
    for x2 in 'КБЛА':
        for x3 in 'КБЛА':
            for x4 in 'КБЛА':
                for x5 in 'КБЛА':
                    for x6 in 'КБЛА':
                        s=x1+x2+x3+x4+x5+x6
                        if s.count('К')==1 and s.count('А')==3 and s.count('Б')==1 and
s.count('Л')==1:
                            if s.count('АА')==0:
                                k=k+1
print(k)
```

Повторяющиеся буквы в строке, из которой берём символы в циклах, всегда убираем.

Слова составляются перестановкой, значит, можно представить, что просто собирают из кубиков **КАБАЛА** различные слова. Следовательно, в наших словах будет ровно одна буква "К", три буквы "А", одна буква "Б" и одна буква "Л". Это программируем с помощью условия и функции **.count()**.

Т.к. повторяется может только буква "А", то мы прописываем условие, что две буквы "А" подряд не могут находиться в наших комбинациях.

Если все условия будут пройдены, то мы подсчитываем такие комбинации.

Ответ: 24

Проанализировать первую или последнюю букву в слове. Узнаем, как это можно сделать с помощью питона.

Задача 4 (Проверяем первую букву слова)

Сколько слов длины 5, начинающихся с гласной буквы, можно составить из букв Е, Г, Э? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.

Решение:

В этой тренировочной задаче из 8 задания **ЕГЭ по информатике 2022** нужно держать на контроле первую букву в слове.

```
k=0
for x1 in 'ЕГЭ':
    for x2 in 'ЕГЭ':
        for x3 in 'ЕГЭ':
            for x4 in 'ЕГЭ':
                for x5 in 'ЕГЭ':
                    s=x1+x2+x3+x4+x5
                    if x1=='Е' or x1=='Э':
                        k=k+1
print(k)
```

Подсчитываем только те комбинации, которые начинаются с гласных букв.

Ответ: 162

Задача 5 (Важный пример)

Сергей составляет 6-буквенные коды из букв С, О, Л, О, В, Е, Ё. Буква Ё может использоваться в коде не более одного раза, при этом она не может стоять на первом месте, на последнем месте и рядом с буквой Е. Все остальные буквы могут встречаться произвольное количество раз или не встречаться совсем. Сколько различных кодов может составить Сергей?

Решение:

Эта задача примечательная тем, что буква "О" в слове "СОЛОВЕЙ" повторяется. В этом случае мы должны убрать повторение буквы из перебора.

```
k=0
for x1 in 'СОЛВЕЙ':
    for x2 in 'СОЛВЕЙ':
        for x3 in 'СОЛВЕЙ':
            for x4 in 'СОЛВЕЙ':
                for x5 in 'СОЛВЕЙ':
                    for x6 in 'СОЛВЕЙ':
                        s=x1+x2+x3+x4+x5+x6
                        if s.count('Й')<=1 and x1!='Й' and x6!='Й' and s.count('ЕЙ')==0 and
s.count('ЙЕ')==0:
                            k=k+1
print(k)
```

Здесь также учитываем остальные условия.

Ответ: 23625

Задача 6 (Количество гласных)

Василий составляет 4-буквенные коды из букв Г, А, Ф, Н, И, Ё. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Ё и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?

Решение:

Напишем программу.

```
k=0
for x1 in 'ГАФНИЙ':
    for x2 in 'ГАФНИЙ':
        for x3 in 'ГАФНИЙ':
            for x4 in 'ГАФНИЙ':
                s=x1+x2+x3+x4
                if x1!='Й' and s.count('А') + s.count('И') >= 1:
                    k=k+1

print(k)
```

Ответ: 888

Задача 7 (Перебор чисел)

Сколько существует чисел, восьмеричная запись которых содержит 7 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Решение:

```
k1=0
k2=0
for x1 in '1234567':
    for x2 in '01234567':
        for x3 in '01234567':
            for x4 in '01234567':
                for x5 in '01234567':
                    for x6 in '01234567':
                        for x7 in '01234567':
                            s=x1+x2+x3+x4+x5+x6+x7
                            if s.count(x1)==1 and s.count(x2)==1 and s.count(x3)==1 and
s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1 and s.count(x7)==1:
                                if int(x1)%2==0 and int(x2)%2==1 and int(x3)%2==0 and
int(x4)%2==1 and int(x5)%2==0 and int(x6)%2==1 and int(x7)%2==0:
                                    k1=k1+1
                                if int(x1)%2==1 and int(x2)%2==0 and int(x3)%2==1 and
int(x4)%2==0 and int(x5)%2==1 and int(x6)%2==0 and int(x7)%2==1:
                                    k2=k2+1

print(k1+k2)
```

Число не может начинаться с нуля. Поэтому ноль был исключён из первого цикла.

Первое условие следит за тем, чтобы каждая цифра встречалась один раз в числе. Второе условие подсчитывает количество вариантов, когда первая цифра чётная. Второе условие следит за тем, чтобы чётность и нечётность цифр чередовалась. Третье условие, наоборот, подсчитывает варианты, когда первая цифра нечётная.

Операция % - остаток от деления. Если остаток от деления на 2 равен нулю, то число чётное. Если остаток от деления на 2 равен 1, то число нечётное.

Функция **int()** преобразует символ в число. Ведь мы работаем именно с символами, а не с реальными числами.

Ответ: 1008

Задача 8 (Числа, Закрепление)

Сколько существует четырёхзначных чисел, записанных в восьмеричной системе счисления, в записи которых ровно две одинаковые цифры, причём стоящие рядом ?

Решение:

k=0

for x1 in '1234567':

for x2 in '01234567':

for x3 in '01234567':

for x4 in '01234567':

if (x1==x2 and x2!=x3 and x2!=x4 and x3!=x4) or (x2==x3 and x3!=x1 and x3!=x4 and x1!=x4) or (x3==x4 and x3!=x2 and x3!=x1 and x1!=x2):

k=k+1

print(k)

Здесь следующий принцип составления условия. Два соседа должны быть равны. Берём одного соседа из пары, где цифры должны быть равны, и комбинируем его с другими цифрами. Пишем уже, чтобы цифры были не равны. Так же прописываем, чтобы две оставшиеся цифры также не были равны. Таким образом, перебираем все варианты.

Ответ: 882

Задача 9 (Числа, важный приём)

Сколько существует различных трёхзначных чисел в шестнадцатеричной системе счисления, в записи которых цифры следуют слева направо в невозрастающем порядке?

Решение:

```
k=0
for x1 in '123456789ABCDEF':
    for x2 in '0123456789ABCDEF':
        for x3 in '0123456789ABCDEF':
            if x1 >= x2 >= x3:
                k=k+1

print(k)
```

Символы можно сравнивать знаками больше или меньше. Символы-цифры сравниваются, как обычные числа. Буквы сравниваются в алфавитном порядке.

Применяем этот приём и получаем ответ.

Ответ: 815

Задача 10 (Две чётные и две нечётные цифры не стоят рядом)

Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 7 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Решение:

```
k=0
for x1 in '123456789':
    for x2 in '0123456789':
        for x3 in '0123456789':
            for x4 in '0123456789':
                for x5 in '0123456789':
                    for x6 in '0123456789':
                        for x7 in '05':

                            s=x1+x2+x3+x4+x5+x6+x7

                            if s.count(x1)==1 and s.count(x2)==1 and s.count(x3)==1 and
s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1 and s.count(x7)==1:
```

```
        if x1 in '02468' and x2 in '13579' and x3 in '02468' and x4 in '13579'
and x5 in '02468' and x6 in '13579' and x7 in '02468':
            k=k+1
        if x1 in '13579' and x2 in '02468' and x3 in '13579' and x4 in '02468'
and x5 in '13579' and x6 in '02468' and x7 in '13579':
            k=k+1

print(k)
```

Перебираем 7 разрядов десятичного числа. С нуля число не может начинаться, поэтому из первого цикла удаляем ноль.

Число должно делиться на 5, значит, в последнем цикле оставляем только 0 и 5.

Все цифры различны, поэтому применяем условие, что символ x1 встречается 1 раз, символ x2 встречается 1 раз и т.д.

Фраза "две чётные и две нечётные цифры не стоят рядом" обозначает, что цифры должны чередоваться. Например, чётная, нечётная, чётная, нечётная и т.д. (или наоборот).

Направление задаёт именно первая цифра, остальные цифры выстраиваются по ней.

Проверить чётность/нечётность цифр просто, мы проверяем существует ли конкретный символ с троке из чётных или нечётных цифр.

У нас два равноправных случая: когда первая цифра чётная, и когда первая цифра нечётная.

Ответ: 2880

Без программирования.

Задача 11(Со списками, классическая)

Все 4-буквенные слова, составленные из букв А, Е, И, О записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААА
2. АААЕ
3. АААИ

4. АААО
5. ААЕА
- ...

Запишите слово, стоящее на 248-м месте от начала списка.

Решение:

Обозначим условно **А** - 0, **Е** - 1, **И** - 2, **О** - 3.

Важно: Нужно буквам присваивать цифры именно в том порядке, в котором они идут в самом правом столбце, потому что буквы могут дать в "перепутанном порядке" (например Е, А, И, О), и тогда ничего не получится.

1. АААА	0
2. АААЕ	1
3. АААИ	2
4. АААО	3
5. ААЕА	
...	

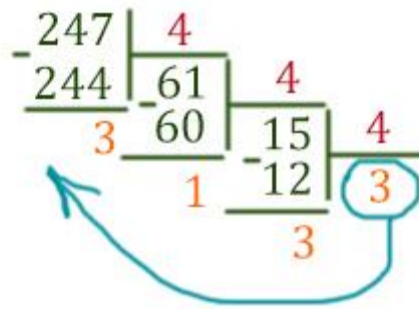
Теперь запишем список с помощью цифр.

1. 0000
2. 0001
3. 0002
4. 0003
5. 0010
- ...

Получился обычный счёт в **четверичной системе!!** (всего используются 4 цифры: 0, 1, 2, 3). А слева нумерация показывает соответствие нашей десятичной системе. Но **все числа десятичной системы** в этой таблице соответствия **сдвинуты на 1**, ведь мы должны были начать с нуля.

Нас просят записать слово стоящее на 248, т.е. если была обычная таблица соответствия чисел десятичной системы и четверичной системы, слово стоящее на 248 месте, находилось бы на 247 (248 - 1) месте. Значит, наше искомое **четверичное число** соответствует 247 в десятичной системе.

Переведём число 247 в четверичную систему!



Получилось число 3313_4 в четверичной системе. Сделаем обратное декодирование в буквы. Таким образом, ответ будет **ООЕО**.

Решение на Python.

```
k=0
for x1 in 'АЕИО':
    for x2 in 'АЕИО':
        for x3 in 'АЕИО':
            for x4 in 'АЕИО':
                s=x1+x2+x3+x4
                k=k+1
                if k==248:
                    print(s)
```

Переменная **k** это порядковый номер слов. Мы подсчитываем все слова без какого-либо условия.

Каждая переменная **x** перебирает буквы, которые используются для составления слов. **Важно:** буквы нужно перебирать в том порядке, в котором они идут **в правом столбце** списка, приведенного в задаче.

Мы составляем 4 вложенных цикла, т.к. длина слов равна 4-м буквам.

В переменной **s** формируется каждое слово из списка. Если порядковый номер равен 248, то печатаем слово **s**.

Ответ: ООЕО

Ещё одна похожая задача 8 задания из примерных вариантов **ЕГЭ по информатике 2022**, но другой вариации.

Задача 12 (Классика, Другая вариация)

Все 5-буквенные слова, составленные из букв А, Р, У, К записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. ААААК
3. ААААР
4. ААААУ
5. АААКА

.....

Укажите номер слова УКАРА

Решение:

Закодируем буквы цифрами: А - 0, К - 1, Р - 2, У - 3. Здесь как раз буквы даны не в том порядке, как они идут в самом правом столбце. Но мы должны кодировать именно в том порядке, как буквы идут в самом правом столбце.

1. ААААА	0
2. ААААК	1
3. ААААР	2
4. ААААУ	3
5. АААКА	

У нас получилось четыре цифры! Значит снова можно слова превратить в таблицу соответствия между **десятичной** системой и **четверичной** системой. Но десятичная система смещена на 1 позицию.

1. 00000
2. 00001
3. 00002
4. 00003
5. 00010

.....

Выписываем данное нам слово и посмотрим, какое число в четверичной системе было бы, если бы у нас были в место слов числа в четверичной системе!

УКАРА

3 1 0 2 0

Получили число в четверичной системе 31020_4 . Узнаем, какое число в десятичной системе соответствовало этому числу, если бы была обычная таблица соответствия. Для этого переведем число 31020_4 из **четверичной** системы в **десятичную**. Перевод делаем по аналогии перевода из двоичной системы в десятичную.

$$0 * 4^0 + 2 * 4^1 + 0 * 4^2 + 1 * 4^3 + 3 * 4^4 = 840 \text{ (в десят. системе)}$$

8 64 768

Но помним, что у нас нумерация идёт на 1 быстрее, нежели мы бы поставили десятичные числа, как в таблице соответствия, потому что нумерация начинается не с нуля, а с 1. Поэтому к числу **840** нужно **прибавить 1**, и в ответе будет **841**

Решение на Python.

```
k=0
for x1 in 'АКРУ':
    for x2 in 'АКРУ':
        for x3 in 'АКРУ':
            for x4 in 'АКРУ':
                for x5 in 'АКРУ':
                    s=x1+x2+x3+x4+x5
                    k=k+1
                    if s=='УКАРА':
                        print(k)
```

Если появилось нужное слово, то печатаем порядковый номер.

Ответ: 841

Задача 13

Все 4-буквенные слова, в составе которых могут быть буквы Н, О, Т, К, И, записаны в алфавитном порядке и пронумерованы, начиная с 1. Ниже приведено начало списка.

1. ИИИИ
2. ИИИК
3. ИИИН
4. ИИИО
5. ИИИТ
6. ИИКИ
- ...

Под каким номером в списке идёт первое слово, которое начинается с буквы О?

Решение:

Закодируем буквы цифрами.



Получилось 5 цифр (0, 1, 2, 3, 4), значит, будем работать в **пятеричной системе**.

Нужно найти номер первого слова, которое начинается с буквы **О**. Если говорить на языке пятеричных чисел, то нужно найти номер числа **3000₅**. Мы "забиваем нулями", чтобы число было четырёхразрядное, т.к. слова 4-х буквенные. Именно нулями, потому что нужно именно первое слово найти.

Теперь, как в предыдущей задаче, переведём число **3000₅** из **пятеричной** системы в **десятичную**.

$$0 * 5^0 + 0 * 5^1 + 0 * 5^2 + 3 * 5^3 = 375 \text{ (в десят. системе)}$$

Но опять же должны прибавить 1 к числу **375**, т.к. нумерация отличается от десятичных чисел на 1 в большую сторону.

Решение на Python.

```
k=0
for x1 in 'ИКНОТ':
    for x2 in 'ИКНОТ':
        for x3 in 'ИКНОТ':
            for x4 in 'ИКНОТ':
                s=x1+x2+x3+x4
                k=k+1
                if x1=='О':
                    print(k)
```

Распечатаются разные номера слов, которые начинаются с буквы "О". Мы выбираем самый первый.

Ответ: 376