



# Кодирование данных, комбинаторика, СИСТЕМЫ СЧИСЛЕНИЯ

**Павлова Инна Борисовна,**  
учитель информатики  
ГБОУ «Инженерная школа № 1581»

*Навигатор самостоятельной подготовки к ЕГЭ-2021*

**ИНФОРМАТИКА и ИКТ**

## **ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ. КОДИРОВАНИЕ ИНФОРМАЦИИ. СИСТЕМЫ СЧИСЛЕНИЯ**

**Какие позиции кодификатора элементов содержания проверяет**

| <b>Код элемента содержания</b> | <b>Элементы содержания, проверяемые заданиями экзаменационной работы</b> | <b>Номер задания</b> |
|--------------------------------|--|----------------------|
| <b>1.1.3</b>                   | Знание о методах измерения количества информации                         | Задание 8            |

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует трёхбуквенные слова, в которых могут быть только буквы Ш, К, О, Л, А, причём буква К появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Ответ: \_\_\_\_\_.

## Актуализация знаний

Сочетания без повторов

$$C_n^m = \frac{n!}{m!(n-m)!}$$

с повторениями

$$\bar{C}_n^m = C_{n+m-1}^m = \frac{(n+m-1)!}{m!(n-1)!}$$

Размещения без повторов

$$A_n^m = \frac{n!}{(n-m)!}$$

с повторениями

$$\bar{A}_n^m = n^m$$

Перестановки без повторов

$$P_n = n!$$

с повторениями

$$\bar{P}_{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! n_2! \dots n_k!}$$

- Ручной перебор
- Комбинаторное решение
- Перебор в программе

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует трёхбуквенные слова, в которых могут быть только буквы Ш, К, О, Л, А, причём буква К появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Ответ: \_\_\_\_\_.

Рассмотрим схему всех возможных вариантов расположения обязательной буквы (К)

$$\begin{array}{ccc} \underline{K} & * & * \\ * & \underline{K} & * \\ * & * & \underline{K} \\ - & - & - \end{array}$$

На каждом из мест, обозначенных «\*» может находиться одна из четырёх букв из набора Ш, О, Л, А.

$$\begin{array}{ccc} \underline{K} & \underline{Ш} & \underline{4} \\ \underline{K} & \underline{О} & \underline{4} \\ \underline{K} & \underline{Л} & \underline{4} \\ \underline{K} & \underline{А} & \underline{4} \end{array}$$

Итого: при первой букве К – 16 вариантов, так как К может стоять на одной из трёх позиций, общее количество подходящих слов –  
 **$3 \cdot 16 = 48$**

- Ручной перебор
- Комбинаторное решение
- Перебор в программе

Число размещений с повторениями:

$$\overline{A}_n^m = n^m$$

Для нашей задачи это означает, что при фиксированном расположении буквы К остальные буквы можно выбрать  $4^2$  различными способами.

**Общее количество решений:  $3 * 4^2 = 48$ .**

- Ручной перебор
- Комбинаторное решение
- **Перебор в программе**

## Решение с помощью программы (Б.С. Михлин)

Решение перебором всех возможных комбинаций символов (a, b, c) и подсчёт среди них комбинаций, где 'к' встречается только один раз.

```
n=0
s='школа'
for a in s:
    for b in s:
        for c in s:
            if (a+b+c).count('к')==1:
                n+=1
print(n)
```

- Ручной перебор
- Комбинаторное решение
- **Перебор в программе**

```
from itertools import product
data = tuple(product('ЛОМ', repeat=3))
n = 0
print(data)
```

```
(('Л', 'Л', 'Л'), ('Л', 'Л', 'О'), ('Л', 'Л', 'М'),
('Л', 'О', 'Л'), ('Л', 'О', 'О'), ('Л', 'О', 'М'),
('Л', 'М', 'Л'), ('Л', 'М', 'О'), ('Л', 'М', 'М'),
('О', 'Л', 'Л'), ('О', 'Л', 'О'), ('О', 'Л', 'М'),
('О', 'О', 'Л'), ('О', 'О', 'О'), ('О', 'О', 'М'),
('О', 'М', 'Л'), ('О', 'М', 'О'), ('О', 'М', 'М'),
('М', 'Л', 'Л'), ('М', 'Л', 'О'), ('М', 'Л', 'М'),
('М', 'О', 'Л'), ('М', 'О', 'О'), ('М', 'О', 'М'),
('М', 'М', 'Л'), ('М', 'М', 'О'), ('М', 'М', 'М'))
>>>
```

Сделаем вывод: пакет itertools содержит функцию product, создающую кортежи из ряда вложенных итерируемых элементов.

- Ручной перебор
- Комбинаторное решение
- Перебор в программе

Для нашей задачи это применимо в следующем алгоритме:

- составляем все кортежи из букв слова «ШКОЛА»;
- считаем количество слов, в которых буква К встречается только один раз.

**Решение с помощью модуля itertools (С.С. Поляков):**

```
from itertools import product
p = product('ШКОЛА', repeat=3)
n = 0
for i in p:
    if i.count('К')==1:
        n+=1
print(n)
```

- Анализ списка слов заданной длины.
- Сколько слов можно составить при заданных ограничениях?
- Сколько чисел можно составить при заданных ограничениях?

155) (А. Минак) Все 6-буквенные слова, составленные из букв А, О, И, Э, У, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. АААААА  
2. АААААИ  
3. АААААО

4. АААААУ

.....

Под каким номером стоит последнее слово, начинающееся и заканчивающееся буквой О?

**Сопоставим буквам, упорядоченным по алфавиту, цифры.**

**Определим систему счисления.**

А – 0; И – 1; О – 2; У – 3; Э – 4. Система счисления пятеричная.

**Составим число, начинающееся на «О» и заканчивающееся на «О», такое, чтобы оно было последним.**

$244442_5$

**Вычислим его значение:**

$\gg\gg 2+4*5+4*5*5+4*5*5*5+4*5*5*5*5+2*5*5*5*5*5$

9372

$\gg\gg$

**Не забудем, что на первом месте стоит число с значением «0».**

Ответ: 9373

## Актуализировать знания:

1. Основание системы счисления
2. Алфавит системы счисления
3. Базис системы счисления
4. Перевод чисел в различные системы счисления

8 Вероника составляет 3-буквенные коды из букв В, Е, Р, О, Н, И, К, А, причём буква В должна входить в код ровно один раз. Все полученные коды Вероника записала в алфавитном порядке и пронумеровала. Начало списка выглядит так:

1. ААВ
2. АВА
3. АВЕ

...

На каком месте будет записан первый код, не содержащий ни одной буквы А?

Ответ: \_\_\_\_\_.

```

('А', 'А', 'В') ('А', 'В', 'А') ('А', 'В', 'Е')
('А', 'В', 'И') ('А', 'В', 'К') ('А', 'В', 'Н')
('А', 'В', 'О') ('А', 'В', 'Р') ('А', 'Е', 'В')
('А', 'И', 'В') ('А', 'К', 'В') ('А', 'Н', 'В')
('А', 'О', 'В') ('А', 'Р', 'В') ('В', 'А', 'А')
('В', 'А', 'Е') ('В', 'А', 'И') ('В', 'А', 'К')
('В', 'А', 'Н') ('В', 'А', 'О') ('В', 'А', 'Р')
('В', 'Е', 'А') ('В', 'Е', 'Е')

```

```

from itertools import product
list_p = list(product('АВЕИКНОР', repeat=3))
n = 0
#print(*list_p)
for i in range(len(list_p)):
    if str(list_p[i]).count('В')==1:
        n+=1
        if str(list_p[i]).count('А')==0:
            print(n)
            break

```

23

>>>

8 Тимофей составляет 5-буквенные коды из букв Т, И, М, О, Ф, Е, Й. Буква Т должна входить в код не менее одного раза, а буква Й – не более одного раза. Сколько различных кодов может составить Тимофей?

## Вручную.

По условию задачи, буква Й может не входить в слово или входить один раз, а буква Т должна быть обязательно.

Если буква Й **не входит ни разу**, то количество слов равно количеству размещений с повторениями шести букв на пяти местах

$$N = 6^5$$

Но это без учёта того, что буква Т обязательно должна быть в слове. Количество слов, в которых нет буквы Т, равно количеству размещений с повторениями пяти букв на пяти местах

$$N_T = 5^5$$

Общее количество слов, в которых нет буквы Й и буква Т встречается хотя бы один раз :

$$N = 6^5 - 5^5$$

Если буква Й **входит один раз**, то количество слов равно количеству размещений с повторениями шести букв на четырёх местах, при условии, что вариантов размещения буквы Й может быть 5.

$$N = 5 * 6^4$$

Но это без учёта того, что буква Т обязательно должна быть в слове. Количество слов, в которых нет буквы Т, равно количеству размещений с повторениями пяти букв на четырёх местах, при условии, что вариантов размещения буквы Й может быть 5.

$$N_T = 5 * 5^4$$

Общее количество: 

```
>>> 6**5-5**5+5*(6**4-5**4)
8006
>>> |
```

```
from itertools import product
n=0
p=product('ТИМОФЕЙ', repeat=5)
for i in p:
    if i.count('Й')<=1 and i.count('Т')>=1:
        n+=1
        #print(i)
print(n)
```

Сколько существует десятичных пятизначных чисел, в которых все цифры различны и никакие две чётные или две нечётные цифры не стоят рядом?

Вручную: рассмотрим две возможные маски числа, удовлетворяющего условию задачи (Ч – чётная цифра, Н – нечётная)

- 1) Ч Н Ч Н Ч
- 2) Н Ч Н Ч Н

Для первой маски количество вариантов рассчитывается  $4 \cdot 5 \cdot 4 \cdot 4 \cdot 3$

Для второй –  $5 \cdot 5 \cdot 4 \cdot 4 \cdot 3$

Ответ задачи: 2160

```
from itertools import permutations
n=0
for i in permutations('0123456789',5):
    if i[0]!='0':
        fl=True
        for j in range(4):
            if (i[j] in '02468' and i[j+1] in '02468') or (i[j] in '13579' and i[j+1] in '13579'):
                fl=False
                break
        if fl:
            n+=1
            #print(''.join(i))
print(n)
```

Саша составляет шестизначные числа, оканчивающиеся на 26, причём цифры в числе не могут повторяться и каждое число содержит ровно три чётные цифры или ровно две нечётные цифры. Сколько различных чисел может составить Саша?

Вручную:

Рассмотрим все возможные маски числа (Ч – чётная цифра, Н – нечётная), удовлетворяющего условию «три чётных»

1) Ч Н Н Н 2 6 –  $2 \cdot 5 \cdot 4 \cdot 3 = 120$

2) Н Ч Н Н 2 6 –  $5 \cdot 3 \cdot 4 \cdot 3 = 180$ , и таких масок будет 3.

Всего при трёх нечётных цифрах получаем  $180 \cdot 3 + 120 = 660$  слов (чисел).

Рассмотрим маски числа, удовлетворяющего условию «два нечётных»

1) Н Н Ч Ч  $5 \cdot 4 \cdot 3 \cdot 2 = 120$

2) Н Ч Н Ч 120

3) Н Ч Ч Н 120

4) Ч Н Н Ч  $2 \cdot 5 \cdot 4 \cdot 2 = 80$

5) Ч Н Ч Н 80

6) Ч Ч Н Н 80

$660 + 120 \cdot 3 + 80 \cdot 3 = 1260$

Ответ задачи: 1260

```
from itertools import permutations as pr
n=0
for i in pr('0123456789',6):
    kcet=2
    knecet=0
    if i[0]!='0':
        for j in range(4):
            if i[4]=="2" and i[5]=="6":
                if (int(i[j]))%2==0:
                    kcet+=1
                if (int(i[j]))%2!=0:
                    knecet+=1
        if knecet==2 or kcet==3:
            n+=1
print(n)
```



```
Python 3.6.2rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2rc1 (heads/3.6:268e1fb, J
ntel)] on win32
Type "copyright", "credits" or "licen
>>>
===== RESTART: C:\Users
1260
>>>
```

## Задание 8

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=F30CA64593C085AE41BEADBE0FB88F62>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=39136552463EB7034E9572B29D3E247F>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=B45FCE8B5DFFBA754EFD1E206B5BE14A>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=C972A0333BB4845D470F6EC0DAA67D4C>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=9B389FD63C92B37B482229923FF8DA95>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=A1D239F30681A9F84B7C6CD04A139065>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=872B802057C2BFA746DEA2FE5915EEF8>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=57C44A7B5FA5B2544982B2DA7B9828B5>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=297D2490140D82ED4BFB7225550B9325>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=25686322709F80BB4F4FC159795E13D8>

<http://ege.fipi.ru/os11/xmodules/qprint/openlogin.php?qst=03EDA029DF43AFE74B7FF1D4ABBBABF1>

Номера с сайта К.Ю. Полякова  
<https://kpolyakov.spb.ru/school/ege.htm>

- Анализ списка слов заданной длины  
№ 1–20; 33–38; 71–74; 78–87; 155;  
182
- Сколько слов можно составить при заданных ограничениях? № 21; 25; 26; 30; 31; 39; 40; 60–67; 69; 70; 89–113; 119; 120; 128–142; 168–174; 178; 181
- Сколько чисел можно составить при заданных ограничениях?  
№156–166





# Кодирование данных, комбинаторика, системы счисления

**Павлова Инна Борисовна,**  
учитель информатики  
ГБОУ «Инженерная школа № 1581»