

8 (базовый уровень, время – 4 мин)

Тема: Кодирование данных, комбинаторика, системы счисления.

Что проверяется:

Знание о методах измерения количества информации (?)

1.6.1. Формализация понятия алгоритма (?)

1.1.4. Читать и отлаживать программы на языке программирования (?)

Что нужно знать:

- В русском языке 33 буквы: 10 гласных букв (а, у, о, ы, и, э, я, ю, ё, е), 21 согласная буква (б, в, г, д, ж, з, й, к, л, м, н, п, р, с, т, ф, х, ц, ч, ш, щ) и два знака (ь, ъ).
- Алфавит английского языка по написанию совпадает с латинским алфавитом и состоит из 26 букв.
- принципы работы с числами, записанными в позиционных системах счисления
- если слово состоит из L букв, причем есть n_1 вариантов выбора первой буквы, n_2 вариантов выбора второй буквы и т.д., то число возможных слов вычисляется как произведение

$$N = n_1 \cdot n_2 \cdot \dots \cdot n_L$$

- если слово состоит из L букв, причем каждая буква может быть выбрана n способами, то число возможных слов вычисляется как $N = n^L$
- если в программе L вложенных циклов и внешний цикл выполняется n_1 раз, следующий (вложенный) n_2 раз и т.д., то команды самого внутреннего цикла будут выполняться N раз, где

$$N = n_1 \cdot n_2 \cdot \dots \cdot n_L.$$

Если $n_1 = n_2 = \dots = n_L = n$, то $N = n^L$.

- при увеличении n или L значение N сильно возрастает, что приводит к существенному увеличению времени выполнения программы.
- (В. Ялдыгин) при решении с помощью программы на языке Python удобно использовать функции из модуля `itertools`:

`combinations` – комбинации, например,

```
from itertools import combinations
cmb = list(combinations('ABC', 2))
print( cmb )
// Результат: [('A', 'B'), ('A', 'C'), ('B', 'C')]
```

`permutations` – перестановки, например,

```
from itertools import permutations
cmb = list(permutations('ABC'))
print( cmb )
// Результат: [('A', 'B', 'C'), ('A', 'C', 'B'),
// ('B', 'A', 'C'), ('B', 'C', 'A'), ('C', 'A', 'B'),
// ('C', 'B', 'A')]
```

`product` – декартово произведение (все возможные слова заданной длины, составленные из данного алфавита), например:

```
from itertools import product
cmb = list(product('ABC', repeat=2))
print( cmb )
// Результат: [('A', 'A'), ('A', 'B'), ('A', 'C'), ('B', 'A'),
// ('B', 'B'), ('B', 'C'), ('C', 'A'), ('C', 'B'), ('C', 'C')]
```

Как видно из этих примеров, результат работы этих трёх функций – массив кортежей. В нём удобно работать с отдельными символами, но неудобно искать сочетания букв. Если нужно работать с сочетаниями букв, нужно «склеить» символы каждого кортежа в строки с помощью метода `.join`:

```

from itertools import product
cmb = product('ABC', repeat=2)
cmb = list( map( "".join, cmb ) )
print( cmb )
// Результат: ['AA', 'AB', 'AC', 'BA', 'BB', 'BC', 'CA',
// 'CB', 'CC']

```

или так:

```

from itertools import product
cmb = product('ABC', repeat=2)
for p in cmb:
    s = "".join(p)
    print( s )
// Результат: AA AB AC BA BB BC CA CB CC

```

- (А. Шпехт) Для перебора всех возможных вариантов в С++ удобно использовать функцию `next_permutation()`, которая генерирует следующую лексикографическую перестановку. Для использования функции необходимо подключить библиотеку `<algorithm>`. Функция переставляет диапазон `[first, last)` в следующую перестановку, причём набор всех перестановок упорядочен лексикографически. Возвращает `true`, если такая «следующая перестановка» существует, и генерирует её; в противном случае преобразует диапазон в лексикографически первую перестановку. Следующий фрагмент выведет «acb»:

```

#include <algorithm>
#include <string>
#include <iostream>
int main()
{
    std::string s = "abc";
    next_permutation(s.begin(), s.end());
    std::cout << s;
}

```

Следующий код печатает все три перестановки строки «aba»:

```

#include <algorithm>
#include <string>
#include <iostream>
int main ( )
{
    std :: string s = "aba" ;
    std :: sort( s.begin() , s.end() ) ;
    do {
        std :: cout << s << '\n' ;
    } while ( std :: next_permutation( s.begin(), s.end() ) ) ;
}

```

Результат:

```

aab
aba
baa

```

Пусть требуется получить все возможные перестановки слова "ЕГЭ". Для вывода в консоль кириллических символов необходимо подключить локаль: `setlocale(LC_ALL, "Russian");`

```

#include <algorithm>
#include <string>
#include <iostream>

```

```
using namespace std;
int main ( )
{
    setlocale(LC_ALL, "Russian");
    string s = "ЕГЭ";
    sort( s.begin(), s.end() );
    do {
        cout << s << '\n' ;
    } while ( next_permutation( s.begin(), s.end() ) ) ;
}
```

Результат:

```
ГЕЭ
ГЭЕ
ЕГЭ
ЕЭГ
ЭГЕ
ЭЕГ
```

Пример задания:

Р-11 (демо-2021). 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?

Решение (теоретическое):

- 1) буква К может стоять на одном из трёх мест, остальные две буквы выбираются из оставшихся четырёх: Ш, О, Л или А
- 2) пусть К – первая буква, тогда оставшиеся две буквы можно выбрать $4^2 = 16$ способами
- 3) так как К может стоять на одной из трёх позиций, общее количество подходящих слов – $3 \cdot 16 = 48$
- 4) Ответ: **48**.

Решение (с помощью программы):

- 1) для проверки решения (при наличии времени) можно использовать рекурсивный перебор (см. учебник К.Ю. Поляков, Е.А. Еремин. Информатика: базовый и углублённый уровни. М.: БИНОМ. Лаборатория знаний, 2019): перебрать всевозможные слова длиной 3 и посчитать те из них, в которых только одна буква К
- 2) шаблон рекурсивной функции, выполняющей перебор, на языке Python может выглядеть так:

```
def rec( word, k, Alpha ):
    global count
    if len(word) == k:
        if valid(word): count += 1
        return
    for c in Alpha:
        rec( word+c, k, Alpha )
```
- 3) функция **rec** принимает три параметра: уже построенную часть слова **word**, заданную длину слов **k** и алфавит **Alpha**
- 4) если длина слова **word** равна **k**, слово полностью построено; проверяем его функцией **valid**, которая возвращает **True**, если слово подходящее;

- 5) если слово удовлетворяет требованиям, увеличиваем глобальную переменную **count** (счётчик подходящих слов)
- 6) если слово ещё не достроено (его длина меньше **k**) в цикле добавляем в конец слова по очереди все буквы из алфавита и вызываем процедуру **rec** рекурсивно
- 7) основная программа выглядит так:

```
count = 0
rec( "", 3, "ШКОЛА" )
print( count )
```

в начальный момент первый аргумент – пустая строка (ещё ни одна буква не выбрана)

- 8) функция **valid** в данной задаче должна проверять, верно ли, что слово содержит ровно одну букву К:

```
def valid( word ):
    return word.count('К') == 1
```

конечно, в другой задаче функцию **valid** нужно изменить в соответствии с условием

- 9) Ответ: 48.

Решение (с помощью программы с функцией):

- 1) можно организовать рекурсивный перебор с помощью функции, которая будет возвращать количество найденных слов:

```
def valid( word ):
    return word.count('К') == 1

def rec( word, k, Alpha ):
    if len(word) == k:
        if valid(word): return 1
        return 0
    count = 0
    for c in Alpha:
        count += rec( word+c, k, Alpha )
    return count

print( rec( "", 3, "ШКОЛА" ) )
```

Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию **product** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

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

- 2) Ответ: 48.

- 3) (Б.С. Михлин) Более компактная запись программы с генератором списка

```
from itertools import product
p=[x for x in product('ШКОЛА', repeat=3)
    if x.count('К')==1 ]
print(len(p))
```

- 4) Ответ: 48.

Решение (с помощью программы, Б.С. Михлин):

- 1) Решение перебором всех возможных комбинаций символов (a, b, c) и подсчет 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)
```

- 2) Ответ: 48.

Решение (с помощью программы на C++, А. Шпехт):

- 1) Наиболее простой вариант решения - комбинация всех вариантов трехсимвольной строки и проверка на количество букв 'К':

```
#include <iostream>
#include <algorithm>
#include <string>
using namespace std;

int main () {
    string st = "ШКОЛА";
    int counter = 0;
    for( char p1: st )
        for( char p2: st )
            for( char p3: st ) {
                string s = "";
                s = s + p1 + p2 + p3;
                if( count(s.begin(), s.end(), 'К') == 1 ) counter++;
            }
    cout << counter;
    return 0;
}
```

- 2) Ответ: 48.

Пример задания:

Р-10. Маша составляет шестибуквенные слова перестановкой букв слова КАПКАН. При этом она избегает слов с двумя подряд одинаковыми буквами. Сколько различных кодов может составить Маша?

Решение:

- 1) если не учитывать, что в слове есть одинаковые буквы, общее количество перестановок 6 букв равно $6! = 720$
- 2) так как перестановка пары одинаковых букв не даёт нового слова, каждая пара уменьшает количество уникальных слов в 2 раза; а у нас 2 пары (повторяются К и А), поэтому количество уникальных слов – в 4 раза меньше, оно равно $720/4 = 180$
- 3) теперь из 180 нужно вычесть количество слов, где встречаются пары КК и АА;
- 4) сначала найдём количество слов, в которых встречаются обе пары, и КК, и АА; обозначим $X=КК$, $Y=АА$, таким образом, нужно найти количество слов из 4-х разных «букв» (Н, П, Х, Y), это количество равно $4! = 24$

- 5) теперь подсчитаем слова, в которых есть Х=КК, но нет АА; получаем набор из 5 «букв» (А, А, Н, П, Х), количество уникальных слов равно $5!/2 = 60$ (учитывая, что перестановка букв А не меняет слово); кроме того, среди них есть еще 24 слова, в которых есть обе пары, то есть имеем $60 - 24 = 36$ слов, где есть КК, но нет АА
- 6) аналогично получаем, что есть **36** слов, где есть АА, но нет КК
- 7) количество нужных нам слов равно $180 - 24 - 36 - 36 = 84$.
- 8) Ответ: **84**.

Решение (с помощью программы, Б.С. Михлин):

- 1) Использование функции `permutations` (перестановки) модуля `itertools` и генератора множества.

```
def ok(x):
    for i in range(5):
        if x[i]==x[i+1]: # соседние буквы д.б. разными
            return False
    else:
        return True

from itertools import permutations
s = 'капкан'
# множество удаляет повторяющиеся перестановки
# (из-за двух 'а' и двух 'к' в s)
m = { x for x in permutations(s) if ok(x) }
print(len(m))
```

- 2) Можно обойтись и без функции (Д. Баянов):

```
from itertools import permutations
s = 'капкан'
m = { x for x in permutations(s)
      if x[0] != x[1] and x[1] != x[2] and x[2] != x[3] and
      x[3] != x[4] and x[4] != x[5] }
print(len(m))
```

- 3) (В.Б. Ялдыгин) Нужно помнить, что функция `permutations` строит не символьные строки, а *кортежи* (неизменяемые списки), поэтому искать в них сочетания букв не получится. Но кортеж символов можно превратить («склеить») в символьную строку с помощью метода `join`, и тогда становится доступен поиск сочетаний, например, так:

```
from itertools import permutations
words = set()
s = 'капкан'
for p in permutations(s):
    w = ''.join(p)
    if w.count('aa') == 0 and w.count('kk') == 0:
        words.add(w)
print(len(words))
```

Здесь учтено, что в слове «КАПКАН» повторяются только буквы К и А, поэтому других парных букв в перестановке появиться не может.

- 4) Ответ: **84**.

Решение (с помощью программы на С++, А. Шпехт):

- 1) Перестановка букв слова "КАПКАН" может дать только 2 пары одинаковых букв - "АА" и "КК". Перебираем все возможные перестановки и проверяем, что в строке нет этих подстрок:

```
#include <iostream>
#include <string>
```

```

#include <algorithm>
using namespace std;
int main(){
    string st = "КАПКАН";
    sort(st.begin(), st.end());
    int counter = 0;
    do{
        if (st.find("AA") == st.npos && st.find("KK") == st.npos)
            counter++;
    } while (next_permutation(st.begin(), st.end()));
    cout << counter;
    return 0;
}

```

2) Ответ: 84.

Ещё пример задания:

Р-09. Маша составляет 5-буквенные коды из букв В, У, А, Л, Ъ. Каждую букву нужно использовать ровно 1 раз, при этом буква Ъ не может стоять на первом месте и перед гласной. Сколько различных кодов может составить Маша?

Решение:

- 1) проще всего сначала найти общее количество возможных слов, а затем вычесть из него количество «запрещённых» слов – тех, которые начинаются на букву Ъ или содержат комбинации ЪУ и ЪА
- 2) сначала найдём общее количество слов, не накладывая никаких ограничений; при этом есть 5 способов выбрать первую букву, 4 способа выбрать вторую и т.д., так что общее число вариантов равно $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$
- 3) первой буквой не может быть Ъ, это исключает $1 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 24$ варианта
- 4) теперь определим, сколько слов содержит запрещённую комбинацию символов ЪУ; эта комбинация может располагаться на одной из 4-х позиций:

$$\text{ЬУ***, *ЬУ**, **ЬУ*, ***ЬУ}$$

первый случай уже исключён (слово не может начинаться с буквы Ъ), для каждого из остальных случаев количество вариантов распределения остальных букв равно $3 \cdot 2 \cdot 1 = 6$ вариантов, то есть запрет сочетания ЪУ исключает $3 \cdot 3 \cdot 2 \cdot 1 = 18$ кодов
- 5) аналогично запрет сочетания ЪА исключает ещё 18 кодов
- 6) таким образом, из 120 слов запрещёнными являются 24 варианта с первой буквой Ъ, 18 вариантов, содержащие ЪУ в середине слова, и 18 вариантов, содержащие ЪА в середине слова
- 7) остаётся $120 - 24 - 18 - 18 = 60$ кодов
- 8) Ответ: 60.

Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию `product` из модуля `itertools`; затем остаётся выбрать и пересчитать подходящие слова:

```

from itertools import product
p = product('ВУАЛЪ', repeat=5)
s = map(lambda x: ''.join(x), p)
n = 0
for x in s:
    if ((x.count('В')==1) and (x.count('У')==1) and
        (x.count('А')==1) and (x.count('Л')==1) and

```

```

        (x.count('Б')==1)) and
        (x[0] != 'Б') and (x.find('БУ')== -1 and
        x.find('БА')== -1):
            n += 1

    print(n)

```

2) Ответ: 60.

Решение (с помощью программы, С. Логинова):

- 1) для построения множества всевозможных слов можно использовать функцию **permutations** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

```

from itertools import permutations
n = 0
for x in permutations('ВУАЛЬ'):
    s = ''.join(x)
    if s[0] != 'Б' and 'БУ' not in s \
        and 'БА' not in s:
        n += 1

print(n)

```

2) Ответ: 60.

Решение (с помощью программы, Б.С. Михлин):

- 5) Можно использовать перебор всех вариантов во вложенном цикле. Когда слово построено, проверяем, подходит ли оно по условию.

```

n=0
s='вуаль'
for a in s[:4]: # s без 'ь'
    for b in s:
        for c in s:
            for d in s:
                for e in s:
                    st=a+b+c+d+e # полученный 5-буквенный код
                    for x in s:
                        if st.count(x)!=1:
                            break
                    else: # все символы встретились ровно один раз
                        if st.count('бу')==0 and st.count('ба')==0:
                            n+=1

print(n)

```

6) Ответ: 60.

Решение (с помощью программы на C++, А. Шпехт):

- 1) Решение:
- ```

#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main(){
 string st = "ВУАЛЬ";
 sort(st.begin(), st.end());
 int counter = 0;
 do{
 if (st[0] != 'Б' && st.find("БУ") == st.npos &&

```



```

 st.find("bA") == st.npos) counter++;
 } while (next_permutation(st.begin(), st.end()));
 cout << counter;
 return 0;
}

```

2) Ответ: **60**.

### Ещё пример задания:

**Р-08.** Вася составляет 4-буквенные коды из букв У, Л, Е, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ЕУ. Сколько различных кодов может составить Вася?

#### Решение:

- 1) проще всего сначала найти общее количество возможных слов, а затем вычесть из него количество слов, в которых есть сочетание ЕУ
- 2) первой буквой не может быть Й, поэтому осталось только 3 возможных первых буквы
- 3) предположим, что первую букву выбрали, тогда вторую выбираем из оставшихся трёх
- 4) при выборе третьей буквы у нас только 2 варианта, а последняя буква – та, которая осталась последней невыбранной:

|   |   |   |   |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
|---|---|---|---|

- 5) в итоге общее количество возможных слов равно  $3 \cdot 3 \cdot 2 \cdot 1 = 18$
- 6) теперь определим, сколько слов содержат сочетание ЕУ; нужно рассмотреть все возможные позиции, где может стоять пара ЕУ
- 7) пусть слово начинается с ЕУ, тогда следующую букву можно выбрать двумя способами, а последнюю – только одним, так что количество вариантов равно 2:

|      |      |   |   |
|------|------|---|---|
| 1(Е) | 1(У) | 2 | 1 |
|------|------|---|---|

- 8) пусть пара ЕУ – это вторая и третья буквы; тогда на первом месте может стоять только буква Л (но не Й), а на последнем – Й, получаем еще один вариант:

|      |      |      |      |
|------|------|------|------|
| 1(Л) | 1(Е) | 1(У) | 1(Й) |
|------|------|------|------|

- 9) сдвиг пары ЕУ в конец слова даёт ещё одну комбинацию

|      |      |      |      |
|------|------|------|------|
| 1(Л) | 1(Й) | 1(Е) | 1(У) |
|------|------|------|------|

- 10) таким образом, из 18 слов четыре ( $2 + 1 + 1$ ) содержат ЕУ

11) Ответ: **14**.

#### Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию **product** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

```

from itertools import product
p = product('УЛЕЙ', repeat=4)
s = map(lambda x: ''.join(x), p)
n = 0
for x in s:
 if (x.count('У') == 1) and (x.count('Л') == 1) and \
 (x.count('Е') == 1) and (x.count('Й') == 1) \
 and (x[0] != 'Й') and (x.find('ЕУ') == -1):
 n += 1
print(n)

```

2) Ответ: **14**.

#### Решение (с помощью программы, С. Логинова):

- 1) для построения множества всевозможных слов можно использовать функцию `permutations` из модуля `itertools`; затем остаётся выбрать и пересчитать подходящие слова:

```
from itertools import permutations
n = 0
for x in permutations('УЛЕЙ', 4):
 s = ''.join(x)
 if (s[0] != 'Й') and (s.count('ЕУ') == 0):
 n += 1
print(n)
```

- 2) Ответ: 14.

Решение (с помощью программы, Б.С. Михлин):

- 1) Можно использовать перебор всех вариантов во вложенном цикле. Когда слово построено, проверяем, подходит ли оно по условию.

```
n=0
s='улей'
for a in s[:3]: # s без 'й'
 for b in s:
 for c in s:
 for d in s:
 st=a+b+c+d # полученный 4-буквенный код
 for x in s:
 if st.count(x)!=1:
 break
 else: # если все символы встретились ровно один раз
 if st.count('еу')==0: # 'еу' - русские буквы
 n+=1
print(n)
```

- 2) Ответ: 14.

Решение (с помощью программы на C++, А. Шпехт):

- 1) В данной задаче необходимо перебрать все комбинации и проверить каждую строку на соответствие условию: нулевой символ не должен быть равен 'Й' (`st[0] != 'Й'`) и в строке не должно встречаться строка "ЕУ" (`st.find("ЕУ") == st.npos`).
- 2) Для перебора всех комбинаций воспользуемся функцией `next_permutation()`. Строку "УЛЕЙ" необходимо предварительно отсортировать в лексикографическом порядке.
- 3) Функция `next_permutation()` возвращает логическое значение существует ли следующая перестановка. Для перебора комбинаций удобней всего воспользоваться циклом с постусловием. Решение:

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main () {
 string st = "УЛЕЙ";
 sort(st.begin(), st.end());
 int counter = 0;
 do {
 if(st[0] != 'Й' && st.find("ЕУ") == st.npos) counter++;
 } while(next_permutation(st.begin(), st.end()));
```

```

 cout << counter;
 return 0;
}

```

4) Ответ: 14.

### Ещё пример задания:

**Р-07.** Вася составляет 3-буквенные слова, в которых есть только буквы В, Е, С, Н, А, причём буква А используется в каждом слове хотя бы 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 Вася?

#### Решение (способ 1):

- 1) буква А может стоять на одном из трёх мест: А\*\*, \*А\*, \*\*А, где \* обозначает любой из пяти символов
- 2) в каждом случае в остальных двух позициях может быть любая из пяти букв
- 3) для шаблона А\*\* получаем (перемножая количество вариантов для каждой позиции)  
 $1 \cdot 5 \cdot 5 = 25$  слов
- 4) для шаблона \*А\* тоже получим 25 слов, но нужно учесть, что все слова, в который первая буква А мы уже подсчитали, поэтому считаем только слова, где на первом месте стоит какая-то другая буква (В, Е, С или Н)
- 5) отсюда находим, что шаблон \*А\* добавляет  $4 \cdot 1 \cdot 5 = 20$  новых слов
- 6) рассматривая шаблон \*\*А, не учитываем уже подсчитанные слова, в которых буква А есть на первом или втором местах, количество новых слов –  $4 \cdot 4 \cdot 1 = 16$
- 7) всего получается  $25 + 20 + 16 = 61$  слово
- 8) Ответ: 61.

#### Решение (способ 2):

- 1) количество слов с буквой А можно вычислить как разность между количеством всех возможных слов и количеством слов, в которых нет буквы А
- 2) количество всех слов  $5 \cdot 5 \cdot 5 = 5^3 = 125$  (на любой из 3-х позиций может стоять любая из 5 букв)
- 3) количество слов, в которых нет буквы А равно  $4 \cdot 4 \cdot 4 = 4^3 = 64$  (на любой из 3-х позиций может стоять любая из 4 букв, кроме А)
- 4) получается  $125 - 64 = 61$  слово, в котором есть буква А (она или несколько)
- 5) Ответ: 61.

#### Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию **product** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

```

from itertools import product
p = product('ВЕСНА', repeat=3)
n = 0
for x in p:
 if (x.count('А') >= 1):
 n += 1
print(n)

```

- 2) Ответ: 61.

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

- 1) можно использовать «метод грубой силы» – перебор всех вариантов:

```

n=0
s='весна'
for a in s:
 for b in s:
 for c in s:
 if (a+b+c).count('a')>=1: # буква 'а' (русская) должна
 # встречаться хотя бы один раз
 n+=1
print(n)

```

2) Ответ: **61**.

Решение (с помощью программы на C++, А. Шпехт):

```

1) Решение:
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main () {
 string s = "ВЕЧА";
 int counter = 0;
 for (char p1: s)
 for (char p2: s)
 for (char p3: s) {
 string st = "";
 st = st + p1 + p2 + p3;
 if (count(st.begin(), st.end(), 'A') >= 1) counter++;
 }
 cout << counter;
 return 0;
}

```

2) Ответ: **61**.

### Ещё пример задания:

**Р-06.** Вася составляет 5-буквенные слова, в которых есть только буквы С, Л, О, Н, причём буква С используется в каждом слове 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 Вася?

Решение:

- 1) буква С может стоять на одном из пяти мест: С\*\*\*\*, \*С\*\*\*, \*\*С\*\*, \*\*\*С\* и \*\*\*\*С, где \* обозначает любой из оставшихся трёх символов
- 2) в каждом случае в остальных четырёх позициях может быть любая из трёх букв Л, О, Н, поэтому при заданном расположении буквы С имеем  $3^4 = 81$  вариант
- 3) всего вариантов  $5 \cdot 81 = 405$ .
- 4) Ответ: **405**.

Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию **product** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

```

from itertools import product
p = product('СЛОН', repeat=5)

```

```

n = 0
for x in p:
 if x.count('C') == 1:
 n += 1
print(n)

```

2) Ответ: 405.

**Решение (с помощью программы, Б.С. Михлин):**

1) можно использовать «метод грубой силы» – перебор всех вариантов:

```

n=0
s='слон'
for a in s:
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 if (a+b+c+d+e).count('с')==1: # буква 'с' (русская)
 # должна встречаться один раз
 n+=1
print(n)

```

2) Ответ: 405.

**Решение (с помощью программы на C++, А. Шпехт):**

1) Решение:

```

#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main () {
 string s = "СЛОН";
 int counter = 0;
 for (char p1: s)
 for (char p2: s)
 for (char p3: s)
 for (char p4: s)
 for (char p5: s) {
 string st = "";
 st = st + p1 + p2 + p3 + p4 + p5;
 if (count(st.begin(), st.end(), 'C') == 1) counter++;
 }
 cout << counter;
 return 0;
}

```

2) Ответ: 405.

### Ещё пример задания:

**Р-05.** Сколько существует различных символьных последовательностей длины 5 в четырёхбуквенном алфавите {A, C, G, T}, которые содержат ровно две буквы A?

**Решение (вариант 1, перебор):**

1) рассмотрим различные варианты слов из 5 букв, которые содержат две буквы A и начинаются с A:

AA\*\*\*

A\*A\*\*

A\*\*A\*

A\*\*\*A

Здесь звёздочка обозначает любой символ из набора {C, G, T}, то есть один из трёх символов.

- 2) итак, в каждом шаблоне есть 3 позиции, каждую из которых можно заполнить тремя способами, поэтому общее число комбинаций (для каждого шаблона!) равно  $3^3 = 27$
- 3) всего 4 шаблона, они дают  $4 \cdot 27 = 108$  комбинаций
- 4) теперь рассматриваем шаблоны, где первая по счёту буква A стоит на второй позиции, их всего три:

\*AA\*\*

\*A\*A\*

\*A\*\*A

они дают  $3 \cdot 27 = 81$  комбинацию

- 5) два шаблона, где первая по счёту буква A стоит на третьей позиции:

\*\*AA\*

\*\*A\*A

они дают  $2 \cdot 27 = 54$  комбинации

- 6) и один шаблон, где сочетание AA стоит в конце

\*\*\*AA

они дают 27 комбинаций

- 7) всего получаем  $(4 + 3 + 2 + 1) \cdot 27 = 270$  комбинаций
- 8) ответ: 270.

#### Решение (вариант 2, использование формул комбинаторики):

- 1) в последовательности из 5 символов нужно использовать ровно две буквы A и три символа, не совпадающих с A, которые обозначим звёздочкой
- 2) сначала найдём количество перестановок из двух букв A и трёх звёздочек
- 3) используем формулу для вычисления числа перестановок с повторениями; для двух разных символов она выглядит так:

$$P(n_A, n_*) = \frac{(n_A + n_*)!}{n_A! n_*!}$$

Здесь  $n_A$  – количество букв A,  $n_*$  – количество звёздочек и восклицательный знак обозначает факториал натурального числа, то есть произведение всех натуральных чисел от 1 до  $n$ :

$$n! = 1 \cdot 2 \cdot \dots \cdot n$$

- 4) в нашем случае  $n_A = 2$  и  $n_* = 3$ , так что получаем

$$P(2,3) = \frac{(2+3)!}{2! 3!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}{1 \cdot 2 \cdot 1 \cdot 2 \cdot 3} = 10$$

- 5) теперь разберёмся со звёздочками: вместо каждой из них может стоять любой из трёх символов (кроме A), то есть на каждую из 10 перестановок мы имеем  $3^3 = 27$  вариантов распределения остальных символов на месте звёздочек
- 6) таким образом, получаем всего  $10 \cdot 27 = 270$  вариантов.
- 7) ответ: 270.

#### Решение (с помощью программы, С.С. Поляков):

- 1) для построения множества всевозможных слов можно использовать функцию `product` из модуля `itertools`; затем остаётся выбрать и пересчитать подходящие слова:

```
from itertools import product
p = product('ACGT', repeat=5)
n = 0
for x in p:
 if (x.count('A') == 2):
 n += 1
print(n)
```

- 2) Ответ: 270.

**Решение (с помощью программы, Б.С. Михлин):**

- 3) можно использовать «метод грубой силы» – перебор всех вариантов:

```
n=0
s='acgt'
for a in s:
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 if (a+b+c+d+e).count('a')==2: # буква 'a' (латинская)
 # должна встречаться два раза
 n+=1
print(n)
```

- 4) Ответ: 270.

**Решение (с помощью программы на C++, А. Шпехт):**

- 1) Наиболее простым решением данной задачи будет решение "в лоб" – перебор всех возможных комбинаций строк и проверка варианта строки на соответствие условию задачи.
- 2) Для подсчета количества символов 'A' в строке удобно использовать функцию `count()` из библиотеки `<algorithm>`:

```
count(first, last, value)
```

`first, last` – итерируемый диапазон, для строки `st` это будет `st.begin(), st.end()`

`value` – проверяемый символ, в нашем случае 'A'.

- 3) Решение:

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main(){
 string s = "ACGT";
 int counter = 0;
 for (char p1: s)
 for (char p2: s)
 for (char p3: s)
 for (char p4: s)
 for (char p5: s){
 string st = "";
 st = st + p1 + p2 + p3 + p4 + p5;
 if (count(st.begin(), st.end(), 'A') == 2) counter++;
 }
 cout << counter;
 return 0;
}
```

- 4) Ответ: 270.

**Ещё пример задания:**

**Р-04.** 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:**

- 1) первая буква слова может быть выбрана двумя способами (Е или Э), остальные – тремя
- 2) общее число различных слов равно  $2 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 162$
- 3) ответ: **162**.

**Решение (через формулы, А.Н. Носкин):**

- 1) Дано слово длиной 5 символов типа **\*\*\*\*\***, где красная звездочка – гласная буква (Е или Э), а черная буква любая из трёх заданных.
- 2) Общая формула количества вариантов:  

$$N = M^L$$
 где  $M$  – мощность алфавита, а  $L$  – длина кода.
- 3) Так как положение одной из букв строго регламентировано (знак умножения в зависимых событиях), то формула всех вариантов примет вид:  $N = M_1^{L_1} \cdot M_2^{L_2}$ ,
- 4) Тогда  $M_1 = 2$  (алфавит гласных букв), а  $L_1 = 1$  (только 1 позиция в слове).  
 $M_2 = 3$  (алфавит всех букв), а  $L_2 = 4$  (оставшиеся 4 позиции в слове).
- 5) В итоге получаем:  $N = 2^1 \cdot 3^4 = 2 \cdot 81 = 162$ .
- 6) ответ: **162**.

**Решение (с помощью программы, С.С. Поляков):**

- 1) для построения множества всевозможных слов можно использовать функцию **product** из модуля **itertools**; затем остаётся выбрать и пересчитать подходящие слова:

```
from itertools import product
p = product('ЕГЭ', repeat=5)
n = 0
for x in p:
 if (x[0] == 'Е') or (x[0] == 'Э'):
 n += 1
print(n)
```

- 2) Ответ: **162**.

**Решение (с помощью программы, Б.С. Михлин):**

- 5) можно использовать «метод грубой силы» – перебор всех вариантов:

```
n=0
s='егэ'
for a in 'еэ': # первая буква должна быть гласной
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 n+=1
print(n)
```

- 6) Ответ: **162**.

**Решение (с помощью программы на C++, А. Шпехт):**

- 1) Для решения задачи можно воспользоваться способом конструирования строки с помощью циклов. Есть два варианта:

**1 способ.** Использование **range-base** циклов, которые будут перебирать строки. Для первой позиции нам необходима строка из гласных букв **st1 = "ЕЭ"**, для позиций 2-5 строка, содержащая все символы **st2 = "ЕГЭ"**. За каждую из 5 позиций будет отвечать отдельный цикл.

```
#include <iostream>
#include <string>
using namespace std;
```



```
int main(){
 string st2 = "ЕГЭ", st1 = "ЕЭ";
 int counter = 0;
 for (auto p1: st1)
 for (auto p2: st2)
 for (auto p3: st2)
 for (auto p4: st2)
 for (auto p5: st2)
 counter++;
 cout << counter;
 return 0;
}
```

- 2) *Примечание.* Не следует использовать строковые литералы:

~~for (auto ch: "abc")~~

Проблема в том, что цикл итерируется по массиву `char ['a', 'b', 'c', '\0']`, а значит цикл выполнится 4 раза. В случае

```
string st = "abc";
for (auto ch: st)
```

цикл итерирует `std::string`, в котором итератор пробежит только по 'a', 'b', 'c' (класс гарантирует завершающий ноль в конце своего буфера, но не итерируется по нему)

- 3) **2 способ.** Произведем замену символов на цифры: Е - 0, Э - 1, Г - 2. Теперь можем перебрать все возможные комбинации. На первой позиции могут стоять только цифры 0 и 1 (гласные буквы). В остальных позициях допустимы цифры 0, 1, 2.

```
#include <iostream>
using namespace std;
int main (){
 int counter = 0;
 for (int p1 = 0; p1 < 2; p1++)
 for (int p2 = 0; p2 < 3; p2++)
 for (int p3 = 0; p3 < 3; p3++)
 for (int p4 = 0; p4 < 3; p4++)
 for (int p5 = 0; p5 < 3; p5++)
 counter++;
 cout << counter;
 return 0;
}
```

- 4) Ответ: **162**.

### Ещё пример задания:

**Р-03.** Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

.....

Запишите слово, которое стоит на 67-м месте от начала списка.

**Решение:**

- 1) самый простой вариант решения этой задачи – использование систем счисления; действительно, здесь расстановка слов в алфавитном порядке равносильна расстановке по возрастанию чисел, записанных в четверичной системе счисления (основание системы счисления равно количеству используемых букв)
- 2) выполним замену  $K \rightarrow 0$ ,  $L \rightarrow 1$ ,  $P \rightarrow 2$ ,  $T \rightarrow 3$ ; поскольку нумерация слов начинается с единицы, а первое число  $KKKK \rightarrow 0000$  равно 0, под номером 67 будет стоять число 66, которое нужно перевести в четверичную систему:  $66 = 1002_4$
- 3) Выполнив обратную замену (цифр на буквы), получаем слово ЛККР.
- 4) Ответ: **ЛККР**.

**Решение (с помощью программы, А.Н. Носкин):**

- 1) на компьютерном ЕГЭ можно использовать программу (язык Python):
 

```
a = ["К", "Л", "Р", "Т"] # буквы К, Л, Р, Т записаны в алфавитном
 # порядке
s = "" # строка для формирования ответа
x = 66 # числовой код слова: 67-1 = 66
while x > 0: # перевод в 4-ю систему счисления
 s += str(x%4)
 x //= 4
s = s[::-1] # реверс строки ответа
for x in s: # формирование СЛОВА
 i = int(x)
 print(a[i], end="")
```
- 2) Ответ: **ЛККР**.

**Решение (с помощью программы, С.С. Поляков):**

- 1) программа на языке Python использует модуль `itertools`:
 

```
from itertools import product
print(*list(product('КЛРТ', repeat=4)) [67-1])
```
- 2) Ответ: **ЛККР**.

**Решение (с помощью программы, Б.С. Михлин):**

- 5) можно использовать «метод грубой силы» – перебор всех вариантов:
 

```
n=0
s='клрт'
for a in s:
 for b in s:
 for c in s:
 for d in s:
 n+=1
 if n==67:
 print(a+b+c+d)
 exit() # выход из Python
```
- 6) другой вариант (с «флажком» для выхода из вложенного цикла) :
 

```
n=0
s='клрт'
fl=False # флажок сброшен
for a in s:
 if fl: break
 for b in s:
 if fl: break
 for c in s:
```

```

if fl: break
for d in s:
 n+=1
 if n==67:
 print(a+b+c+d)
 fl=True # флажок установлен для выхода
 break

```

7) Ответ: **ЛККР**.

### Ещё пример задания:

**P-02.** Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на 240-м месте от начала списка.

#### Решение (1 способ, перебор с конца):

- 1) подсчитаем, сколько всего 5-буквенных слов можно составить из трех букв;
- 2) очевидно, что есть всего 3 однобуквенных слова (А, О, У); двух буквенных слов уже  $3 \times 3 = 9$  (АА, АО, АУ, ОА, ОО, ОУ, УА, УО и УУ)
- 3) аналогично можно показать, что есть всего  $3^5 = 243$  слова из 5 букв
- 4) очевидно, что последнее, 243-е слово – это УУУУУ
- 5) далее идём назад: предпоследнее слово УУУУО (242-е), затем идет УУУУА (241-е) и, наконец, УУУОУ (240-е)
- 6) Ответ: **УУУОУ**.

#### Возможные ловушки и проблемы:

- хорошо, что требовалось найти слово, которое стоит близко к концу списка; если бы было нужно, скажем, 123-е слово, работы было бы значительно больше

#### Решение (2 способ, троичная система, идея М. Густокашина):

- 1) по условию задачи важно только то, что используется набор из трех разных символов, для которых задан порядок (алфавитный); поэтому для вычислений можно использовать три любые символа, например, цифры 0, 1 и 2 (для них порядок очевиден – по возрастанию)
- 2) выпишем начало списка, заменив буквы на цифры:
 

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

.....
- 3) это напоминает (в самом деле, так оно и есть!) числа, записанные в троичной системе счисления в порядке возрастания: на первом месте стоит число 0, на втором – 1 и т.д.
- 4) тогда легко понять, что 240-м месте стоит число 239, записанное в троичной системе счисления
- 5) переведем 239 в троичную систему:  $239 = 22212_3$
- 6) заменяем обратно цифры на буквы:  $22212 \rightarrow \text{УУУОУ}$
- 7) Ответ: **УУУОУ**.

**Возможные ловушки и проблемы:**

- нужно помнить, что нумерация в задаче начинается с 1, а числа в троичной системе – с нуля, поэтому для получения 240-го элемента списка нужно переводить в троичную систему число  $240-1 = 239$ .

**Решение (3 способ, закономерности в чередовании букв, И.Б. Курбанова):**

- 3) подсчитаем, сколько всего 5-буквенных слов можно составить из трех букв:

$3^5 = 243$  слова; 240-ое место – четвертое с конца;

- 4) так как слова стоят в алфавитном порядке, то первая треть (81 шт) начинаются с «А», вторая треть (тоже 81) – с «О», а последняя треть – с «У», то есть первая буква меняется через 81 слово

- 5) аналогично:

- 2-я буква меняется через  $81/3 = 27$  слов;
- 3-я буква – через  $27/3 = 9$  слов;
- 4-я буква – через  $9/3 = 3$  слова и
- 5-я буква меняется в каждой строке.

- 6) из этой закономерности ясно, что

- на первой позиции в искомом слове будет буква «У» (последние 81 букв);
- на второй – тоже буква «У» (последние 27 букв);
- на третьей – тоже буква «У» (последние 9 букв);
- на четвертой – буква «О» (т.к. последние три буквы «У», а перед ними 3 буквы «О»);
- на пятой – буква «У» (т.к. последние 3 буквы чередуются «А», «О», «У», а перед ними такая же последовательность).

- 7) Ответ: **УУУОУ**.

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| 1   | А   | А   | А   | А   | А   |
| 2   | А   | А   | А   | А   | О   |
| 3   | А   | А   | А   | А   | У   |
| 4   | А   | А   | А   | О   | А   |
| ... | ... | ... | ... | ... | ... |
| ... |     |     |     |     |     |
| ... |     |     |     |     |     |
| 240 | У   | У   | У   | О   | У   |
| 241 | У   | У   | У   | У   | А   |
| 242 | У   | У   | У   | У   | О   |
| 243 | У   | У   | У   | У   | У   |

**Решение (с помощью программы, А.Н. Носкин):**

- 1) на компьютерном ЕГЭ можно использовать программу (язык Python):

```
a = ["А", "О", "У"] # буквы А, О, У в алфавитном порядке
s = "" # строка для формирования ответа
x = 239 # числовой код слова: 240-1 = 239
while x > 0: # перевод в троичную систему счисления
 s += str(x%3)
 x //= 3
s = s[::-1] # реверс строки ответа
for x in s: # формирование СЛОВА
 i = int(x)
 print(a[i], end="")
```

- 2) Ответ: **УУУОУ**.

**Решение (с помощью программы, С.С. Поляков):**

- 1) программа на языке Python использует модуль `itertools`:

```
from itertools import product
print(*list(product('АОУ', repeat=5)) [240-1])
```

- 2) Ответ: **УУУОУ**.

**Решение (с помощью программы, Б.С. Михлин):**

- 8) можно использовать «метод грубой силы» – перебор всех вариантов:

```
from sys import exit
n=0
```

```

s='aoy'
for a in s:
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 n+=1
 if n==240:
 print(a+b+c+d+e)
 exit() # выход из Python

```

9) Ответ: **уууоу**.

### Еще пример задания (автор – В.В. Путилов):

**P-01.** Все 5-буквенные слова, составленные из 5 букв А, К, Л, О, Ш, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААК
3. ААААЛ
4. ААААО
5. ААААШ
6. АААКА

.....

На каком месте от начала списка стоит слово ШКОЛА?

#### Решение:

- 1) по аналогии с предыдущим решением будем использовать пятеричную систему счисления с заменой  $A \rightarrow 0$ ,  $K \rightarrow 1$ ,  $L \rightarrow 2$ ,  $O \rightarrow 3$  и  $Sh \rightarrow 4$
- 2) слово ШКОЛА запишется в новом коде так:  $41320_5$
- 3) переводим это число в десятичную систему:  

$$41320_5 = 4 \cdot 5^4 + 1 \cdot 5^3 + 3 \cdot 5^2 + 2 \cdot 5^1 = 2710$$
- 4) поскольку нумерация элементов списка начинается с 1, а числа в пятеричной системе – с нуля, к полученному результату нужно прибавить 1, тогда...
- 5) Ответ: **2711**.

#### Возможные ловушки и проблемы:

- нужно помнить, что список в задании начинается с 1, а числа в троичной системе – с нуля, поэтому для получения N-ой по счёту цепочки нужно переводить в троичную систему число N-1.

#### Решение (с помощью программы, С.С. Поляков):

- 10) программа на языке Python использует модуль `itertools`:

```

from itertools import product
s = list(map(lambda x: ''.join(x), product('АКЛОШ', repeat=5)))
print(s.index('ШКОЛА')+1)

```

- 11) Ответ: **2711**.

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

- 1) можно использовать «метод грубой силы» – перебор всех вариантов:

```

from sys import exit
n=0
s='аклош'

```

```

for a in s:
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 n+=1
 print(n,a+b+c+d+e) # отладочная
 if a+b+c+d+e=='школа':
 print(n)
 exit() # выход из Python

```

2) Ответ: 2711.

### Еще пример задания:

**P-00.** Все 5-буквенные слова, составленные из букв А, О, У, записаны в **обратном** алфавитном порядке. Вот начало списка:

1. УУУУУ
2. УУУУО
3. УУУУА
4. УУУОУ

.....

Запишите слово, которое стоит на 240-м месте от начала списка.

#### Решение (2 способ, троичная система, идея М. Густокашина):

- 1) по условию задачи важно только то, что используется набор из трех разных символов, для которых задан порядок (алфавитный); поэтому для вычислений можно использовать три любые символа, например, цифры 0, 1 и 2 (для них порядок очевиден – по возрастанию)
- 2) выпишем начало списка, заменив буквы на цифры так, чтобы **порядок символов был обратный алфавитный** ( $У \rightarrow 0$ ,  $О \rightarrow 1$ ,  $А \rightarrow 2$ ):

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

.....

- 3) это напоминает (в самом деле, так оно и есть!) числа, записанные в троичной системе счисления в порядке возрастания: на первом месте стоит число 0, на втором – 1 и т.д.
- 4) тогда легко понять, что 240-м месте стоит число 239, записанное в троичной системе счисления
- 5) переведем 239 в троичную систему:  $239 = 22212_3$
- 6) заменяем обратно цифры на буквы, **учитывая обратный алфавитный порядок** ( $0 \rightarrow У$ ,  $1 \rightarrow О$ ,  $2 \rightarrow А$ ):  $22212 \rightarrow АААОА$
- 7) Ответ: АААОА.

#### Решение (с помощью программы, С.С. Поляков):

- 1) программа на языке Python использует модуль `itertools`:
 

```

from itertools import product
print(*list(product('УОА', repeat=5)) [240-1])

```
- 2) Ответ: АААОА.

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

- 1) можно использовать «метод грубой силы» – перебор всех вариантов:
 

```

from sys import exit
n=0

```

```
s='yoa' # порядок символов, как в списке условия задачи
for a in s:
 for b in s:
 for c in s:
 for d in s:
 for e in s:
 n+=1
 if n==240:
 print(a+b+c+d+e)
 exit() # выход из Python
```

2) Ответ: АААОА.

**Задачи для тренировки<sup>1</sup>:**

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

1 . ААААА  
 2 . ААААО  
 3 . ААААУ  
 4 . АААОА  
 .....

Запишите слово, которое стоит на 101-м месте от начала списка.

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

1 . ААААА  
 2 . ААААО  
 3 . ААААУ  
 4 . АААОА  
 .....

Запишите слово, которое стоит на 125-м месте от начала списка.

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

1 . ААААА  
 2 . ААААО  
 3 . ААААУ  
 4 . АААОА  
 .....

Запишите слово, которое стоит на 170-м месте от начала списка.

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

1 . ААААА  
 2 . ААААО  
 3 . ААААУ  
 4 . АААОА  
 .....

Запишите слово, которое стоит на 210-м месте от начала списка.

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

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

Запишите слово, которое стоит на 150-м месте от начала списка.

<sup>1</sup> Источники заданий:

1. Демонстрационные варианты КИМ ЕГЭ 2004-2016 гг.
2. Тренировочные работы МИОО.
3. Крылов С.С., Ушаков Д.М. ЕГЭ 2015. Информатика. Тематические тестовые задания. — М.: Экзамен, 2015.
4. Ушаков Д.М. ЕГЭ-2015. Информатика. 20 типовых вариантов экзаменационных работ для подготовки к ЕГЭ. — М.: Астрель, 2014.



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

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

Запишите слово, которое стоит на 250-м месте от начала списка.

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

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

Запишите слово, которое стоит на 350-м месте от начала списка.

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

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

Запишите слово, которое стоит на 450-м месте от начала списка.

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

1. ААААА  
2. ААААО  
3. ААААУ  
4. АААОА  
.....

Укажите номер первого слова, которое начинается с буквы У.

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

1. ААААА  
2. ААААО  
3. ААААУ  
4. АААОА  
.....

Укажите номер слова ОАОАО.

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

1. ААААА  
2. ААААО  
3. ААААУ  
4. АААОА  
.....

Укажите номер слова УАУАУ.

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

1. ААААА  
2. ААААО  
3. ААААУ  
4. АААОА  
.....

Укажите номер первого слова, которое начинается с буквы О.

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

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

Укажите номер первого слова, которое начинается с буквы У.

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

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

Укажите номер первого слова, которое начинается с буквы К.

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

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

Укажите номер слова РУКАА.

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

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

Укажите номер слова УКАРА.

- 17) Все 5-буквенные слова, составленные из букв К, О, Р, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ККККК  
2. ККККО  
3. ККККР  
4. КККОК  
.....

Запишите слово, которое стоит под номером **238**.

- 18) Все 5-буквенные слова, составленные из букв И, О, У, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1 . ИИИИИ  
2 . ИИИИО  
3 . ИИИИУ  
4 . ИИИОИ  
.....

Запишите слово, которое стоит под номером **240**.

- 19) Все 4-буквенные слова, составленные из букв М, А, Р, Т, записаны в алфавитном порядке. Вот начало списка:

1 . АААА  
2 . АААМ  
3 . АААР  
4 . АААТ  
.....

Запишите слово, которое стоит на **250**-м месте от начала списка.

- 20) Все 5-буквенные слова, составленные из букв Р, О, К, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1 . ККККК  
2 . ККККО  
3 . ККККР  
4 . КККОК  
.....

Запишите слово, которое стоит под номером **182**.

- 21) Сколько слов длины 4, начинающихся с согласной буквы, можно составить из букв Л, Е, Т, О? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.
- 22) Сколько существует различных символьных последовательностей длины 5 в трёхбуквенном алфавите {К, О, Т}, которые содержат ровно две буквы О?
- 23) Сколько существует различных символьных последовательностей длины 6 в трёхбуквенном алфавите {К, О, Т}, которые содержат ровно две буквы К?
- 24) Сколько существует различных символьных последовательностей длины 6 в четырёхбуквенном алфавите {М, А, Р, Т}, которые содержат ровно две буквы Р?
- 25) Сколько слов длины 6, начинающихся с согласной буквы, можно составить из букв Т, О, К? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.
- 26) Сколько слов длины 5, начинающихся с согласной буквы и заканчивающихся гласной буквой, можно составить из букв К, У, М, А? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.
- 27) Вася составляет 6-буквенные слова, в которых есть только буквы К, Р, О, Т, причём буква О используется в каждом слове 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 Вася?
- 28) Вася составляет 5-буквенные слова, в которых есть только буквы К, Р, А, Н, Т, причём буква К используется в каждом слове ровно 2 раза. 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 Вася?

- 29) Вася составляет 6-буквенные слова, в которых есть только буквы К, А, Н, Т, причём буква К используется в каждом слове ровно 2 раза. 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 Вася?
- 30) Сколько слов длины 6, начинающихся и заканчивающихся согласной буквой, можно составить из букв Г, О, Д? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.
- 31) Сколько слов длины 4, начинающихся с согласной буквы и заканчивающихся гласной буквой, можно составить из букв М, Е, Т, Р, О? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.
- 32) (**Е.В. Хламов**) Сколько существует различных символьных последовательностей длины 3 в четырёхбуквенном алфавите {А,В,С,Д}, если известно, что одним из соседей А обязательно является Д, а буквы В и С никогда не соседствуют друг с другом?
- 33) (**А.Н. Носкин**) Все 5-буквенные слова, составленные из букв П, О, Р, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ООООО  
 2. ООООП  
 3. ООООР  
 4. ООООТ  
 5. ОООПО  
 .....

Какое количество слов находятся между словами ТОПОР и РОПОТ (включая эти слова)?

- 34) Все 5-буквенные слова, составленные из букв А, З, Н, С, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ААААА  
 2. ААААЗ  
 3. ААААН  
 4. ААААС  
 5. АААЗА  
 .....

Какое количество слов находятся между словами САЗАН и ЗАНАС (включая эти слова)?

- 35) Все 5-буквенные слова, составленные из букв Д, К, М, О, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ДДДДД  
 2. ДДДДК  
 3. ДДДДМ  
 4. ДДДДО  
 5. ДДДКД  
 .....

Какое количество слов находятся между словами ДОМОК и КОМОД (включая эти слова)?

- 36) Все 4-буквенные слова, составленные из букв М, А, Р, Т, записаны в алфавитном порядке. Вот начало списка:

1. АААА  
 2. АААМ  
 3. АААР  
 4. АААТ  
 .....

Какое количество слов находятся между словами МАРТ и РАМТ (включая эти слова)?

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

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

Какое количество слов находятся между словами РУКАА и УКАРА (включая эти слова)?

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

1. ААААА  
2. ААААО  
3. ААААУ  
4. АААОА  
.....

Какое количество слов находятся между словами УАУАУ и ОУОУА (включая эти слова)?

- 39) Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 4-буквенные слова, в которых есть только буквы А, В, С, D, X, причём буква X 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?
- 40) Алексей составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Алексей использует 5-буквенные слова, в которых есть только буквы А, В, С, X, причём буква X может появиться на последнем месте или не появиться вовсе. Сколько различных кодовых слов может использовать Алексей?
- 41) Вася составляет 3-буквенные слова, в которых есть только буквы К, Р, А, Н, причём буква А используется в каждом слове хотя бы 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 Вася?
- 42) Вася составляет 4-буквенные слова, в которых есть только буквы Л, Е, Т, О, причём буква Е используется в каждом слове хотя бы 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 Вася?
- 43) Вася составляет 5-буквенные слова, в которых есть только буквы Л, Е, Т, О, причём буква Е используется в каждом слове хотя бы 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 Вася?
- 44) Вася составляет 4-буквенные слова, в которых есть только буквы К, Л, О, У, Н, причём буква У используется в каждом слове хотя бы 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 Вася?
- 45) Вася составляет 5-буквенные слова, в которых есть только буквы К, Л, О, У, Н, причём буква У используется в каждом слове хотя бы 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая

Вася составляет 3-буквенные слова, в которых есть только буквы Б, А, Л, К, О, Н, причём буква Б используется в каждом слове хотя бы 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 Вася?

48) Вася составляет 5-буквенные слова, в которых есть только буквы Б, А, Л, К, О, Н, причём буква Б используется в каждом слове хотя бы 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 Вася?

50) Вася составляет 4-буквенные слова, в которых есть только буквы К, А, Т, Е, Р, причём буква Р используется в каждом слове хотя бы 2 раза. 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 Вася?

52) (М.В. Кузнецова) Вася составляет 4-буквенные слова, в которых есть только буквы К, О, М, А, Р, причём буква А используется в них не более 3-х раз (или не используется совсем). 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 Вася?

54) (М.В. Кузнецова) Вася составляет 5-буквенные слова, в которых есть только буквы С, Л, О, Н, причём в каждом слове используется буква О, но не более 3-х 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 Вася?

- 55) (**М.В. Кузнецова**) Вася составляет 6-буквенные слова, в которых есть только буквы Ж, И, Р, А, Ф, причём в каждом слове используется буква А, но не более 4-х 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 Вася?
- 56) (**М.В. Кузнецова**) Вася составляет 5-буквенные слова, в которых есть только буквы С, И, Р, О, П, причём в каждом слове обязательно есть ровно одна буква О, при этом стоять она может только после согласной. 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 Вася?
- 57) (**М.В. Кузнецова**) Вася составляет 6-буквенные слова, в которых есть только буквы П, И, Р, О, Г, причём в каждом слове есть ровно одна буква Р, при этом после неё обязательно стоит гласная буква. 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 Вася?
- 58) (**М.В. Кузнецова**) Вася составляет 4-буквенные слова, в которых есть только буквы П, И, Р, О, Г, причём в каждом слове буква О может встречаться не более двух раз, при этом, если она есть, то перед ней обязательно стоит согласная буква. 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 Вася?
- 59) (**М.В. Кузнецова**) Вася составляет 5-буквенные слова, в которых есть только буквы П, И, Р, О, Г, причём в каждом слове буква Р может встречаться не более двух раз, при этом, если она есть, то после неё обязательно стоит гласная буква. Все допустимые буквы, кроме Р, могут 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 Вася?
- 60) Иван составляет 5-буквенные слова из букв А, Б, В, Г, Э, Ю, Я. Первой и последней буквами этого слова могут быть только буквы Э, Ю или Я, на остальных позициях эти буквы не встречаются. Сколько различных кодовых слов может составить Иван?
- 61) Иван составляет 5-буквенные слова из букв А, Б, В, Г, Д, Э, Ю, Я. Первой и последней буквами этого слова могут быть только буквы Э, Ю или Я, на остальных позициях эти буквы не встречаются. Сколько различных кодовых слов может составить Иван?
- 62) Иван составляет 3-буквенные слова из букв А, Б, В, Г, Д, Я. Буква Я в слове может быть только одна (или ни одной) и только на первой или последней позициях. Сколько различных кодовых слов может составить Иван?
- 63) Иван составляет 4-буквенные слова из букв А, Б, В, Г, Д, Я. Буква Я в слове может быть только одна (или ни одной) и только на первой или последней позициях. Сколько различных кодовых слов может составить Иван?
- 64) Иван составляет 4-буквенные слова из букв А, Б, В, Г, Д, Я. В каждом слове содержится ровно одна буква Я, причём только на первой или последней позициях. Сколько различных кодовых слов может составить Иван?
- 65) Иван составляет 5-буквенные слова из букв А, Б, В, Г, Д, Я. В каждом слове содержится ровно одна буква Я, причём только на первой или последней позициях. Сколько различных кодовых слов может составить Иван?
- 66) (прислал **А.Н. Носкин**) Палиндром – это символьная строка, которая читается одинаково в обоих направлениях. Сколько различных 4-символьных палиндромов можно составить из строчных латинских букв? (В латинском алфавите 26 букв).



- 67) (прислал **А.Н. Носкин**) Палиндром – это символьная строка, которая читается одинаково в обоих направлениях. Сколько различных 6-символьных палиндромов можно составить из строчных латинских букв? (В латинском алфавите 26 букв).
- 68) (**Д.В. Богданов**) Сколько существует способов разместить на книжной полке шесть книг, среди которых имеются четыре тома романа «Война и мир», которые должны стоять рядом (но не обязательно по порядку)?
- 69) Олег составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Олег использует 4-буквенные слова, в которых есть только буквы А, Б, В, Г, Д и Е, причём буква Г появляется ровно 1 раз и только на первом или последнем месте. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Олег?
- 70) Олег составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Олег использует 5-буквенные слова, в которых есть только буквы А, Б, В, и Г, причём буква Г появляется не более одного раза и только на последнем месте. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Олег?
- 71) Все 3-буквенные слова, составленные из букв У, Ч, Е, Н, И, К записаны в алфавитном порядке. Вот начало списка:

1. **ЕЕЕ**  
 2. **ЕЕИ**  
 3. **ЕЕК**  
 4. **ЕЕН**  
 5. **ЕЕУ**  
 .....

Запишите номер первого слова, которое начинается на букву К.

- 72) Все 4-буквенные слова, составленные из букв Ш, К, О, Л, А записаны в алфавитном порядке. Вот начало списка:

1. **АААА**  
 2. **АААК**  
 3. **АААЛ**  
 4. **АААО**  
 5. **АААШ**  
 .....

Запишите номер первого слова, которое начинается на букву О.

- 73) Все 5-буквенные слова, составленные из букв Р, А, Ф, Т записаны в алфавитном порядке. Вот начало списка:

1. **ААААА**  
 2. **ААААР**  
 3. **ААААТ**  
 4. **ААААФ**  
 5. **АААРА**  
 .....

Запишите номер первого слова, которое начинается на букву Т.

- 74) Все 6-буквенные слова, составленные из букв Д, А, Р записаны в алфавитном порядке. Вот начало списка:

1. **АААААА**  
 2. **АААААД**  
 3. **АААААР**



4 . ААААДА

5 . ААААДД

.....

Запишите номер первого слова, которое начинается на букву Р.

- 75) (А.Н. Носкин, г. Москва) Дано слово КОРАБЛИК. Таня решила составлять новые 6-буквенные слова из букв этого слова по следующим правилам: 1) слово начинается с согласной буквы; 2) согласные и гласные буквы в слове должны чередоваться; 3) буквы в слове не должны повторяться. Сколько существует таких слов?
- 76) (А.Н. Носкин, г. Москва) Дано слово КОРАБЛИКИ. Таня решила составлять новые 5-буквенные слова из букв этого слова по следующим правилам: 1) слово начинается с гласной буквы; 2) гласные и согласные буквы в слове должны чередоваться; 3) буквы в слове не должны повторяться. Сколько существует таких слов?
- 77) (А.Н. Носкин, г. Москва) Дано слово ТАРАКАНИЩЕ. Таня решила составлять новые 6-буквенные слова из букв этого слова по следующим правилам: 1) слово начинается с согласной буквы; 2) согласные и гласные буквы в слове должны чередоваться; 3) буквы в слове не должны повторяться. Сколько существует таких слов?
- 78) (Досрочный ЕГЭ-2018) Все 4-буквенные слова, составленные из букв А, И, О, У, Э, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1 . АААА

2 . АААИ

3 . АААО

4 . АААУ

.....

Под каким номером стоит слово ИААЭ?

- 79) Все четырёхбуквенные слова, составленные из букв Г, О, Р, А, записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

1 . АААА

2 . АААГ

3 . АААО

4 . АААР

5 . ААГА

...

Под каким номером в списке идёт первое слово, в котором нет буквы А?

- 80) Все четырёхбуквенные слова, составленные из букв В, Е, Г, А, Н записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

1 . АААА

2 . АААВ

3 . АААГ

4 . АААЕ

5 . АААН

6 . ААВА

...

Под каким номером в списке идёт первое слово, в котором нет буквы А?

- 81) Все четырёхбуквенные слова, составленные из букв В, И, Н, О, Г, Р, А, Д записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

1 . АААА

2 . АААВ

3 . АААГ

4 . АААД

5 . АААИ

- 6. АААН
- 7. АААО
- 8. АААР
- 9. ААВА
- ...

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

- 82) Все четырёхбуквенные слова, составленные из букв В, И, Н, О, Г, Р, А, Д записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

- 1. АААА
- 2. АААВ
- 3. АААГ
- 4. АААД
- 5. АААИ
- 6. АААН
- 7. АААО
- 8. АААР
- 9. ААВА
- ...

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

- 83) Все четырёхбуквенные слова, составленные из букв А, Л, Г, О, Р, И, Т, М записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

- 1. АААА
- 2. АААГ
- 3. АААИ
- 4. АААЛ
- 5. АААМ
- 6. АААО
- 7. АААР
- 8. АААТ
- 9. ААГА
- ...

Под каким номером в списке идёт последнее слово, которое заканчивается на АЛ?

- 84) Все четырёхбуквенные слова, составленные из букв А, Л, Г, О, Р, И, Т, М записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

- 1. АААА
- 2. АААГ
- 3. АААИ
- 4. АААЛ
- 5. АААМ
- 6. АААО
- 7. АААР
- 8. АААТ
- 9. ААГА
- ...

Под каким номером в списке идёт последнее слово, которое заканчивается на ИМ?

- 85) (А.Н. Носкин) Все пятибуквенные слова, составленные из букв У, А, О, записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

- 1. ААААА
- 2. ААААО
- 3. ААААУ
- 4. АААОА

...

Под каким номером в списке идёт первое слово, в котором средняя (третья по счёту) буква – У?

- 86) (А.Н. Носкин) Все пятибуквенные слова, составленные из букв В, Е, Н, О, К, записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

1. ВВВВВ
2. ВВВВЕ
3. ВВВВК
4. ВВВВН
5. ВВВВО
6. ВВВЕВ

...

Под каким номером в списке идёт последнее слово, в котором буквы О и Е встречаются по одному разу?

- 87) (А.Н. Носкин) Все пятибуквенные слова, составленные из букв В, Е, Н, О, К, записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка выглядит так:

1. ВВВВВ
2. ВВВВЕ
3. ВВВВК
4. ВВВВН
5. ВВВВО
6. ВВВЕВ

...

Под каким номером в списке идёт последнее слово, в котором буквы Н и К встречаются ровно по два раза?

- 88) (А.Н. Носкин) Пётр составляет слова длиной 6 букв, которые начинаются с гласной буквы. Всего он смог составить 486 комбинаций слов. Сколько согласных букв использует Петр для составления слов, если известно, что в используемом алфавите только две гласные? Каждая буква может входить в слово несколько раз.
- 89) (А.Н. Носкин) Пётр составляет слова длиной 6 букв, которые начинаются с гласной буквы и всегда заканчиваются согласной буквой Г. Всего он смог составить 512 комбинаций слов. Сколько различных согласных букв использует Петр для составления слов, если известно, что в используемом алфавите только две гласные? Каждая буква может входить в слово несколько раз.
- 90) Вася составляет 4-буквенные коды из букв К, Р, О, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ОЙ. Сколько различных кодов может составить Вася?
- 91) Вася составляет 5-буквенные коды из букв К, А, Л, И, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ИА. Сколько различных кодов может составить Вася?
- 92) Вася составляет 5-буквенные коды из букв М, А, Н, О, К. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы О и не может содержать сочетания АО. Сколько различных кодов может составить Вася?
- 93) Вася составляет 6-буквенные коды из букв П, А, Й, Щ, И, К. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ИА. Сколько различных кодов может составить Вася?
- 94) Вася составляет 6-буквенные коды из букв П, А, Н, Е, Л, Ь. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ь и не может содержать сочетания ЕЬ. Сколько различных кодов может составить Вася?

- 95) Вася составляет 7-буквенные коды из букв К, А, Б, И, Н, Е, Т. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Б и не может содержать сочетания ЕА. Сколько различных кодов может составить Вася?
- 96) Вася составляет 7-буквенные коды из букв К, О, М, Б, А, Й, Н. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания АЙ. Сколько различных кодов может составить Вася?
- 97) Вася составляет 5-буквенные коды из букв К, А, Л, И, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ИАК. Сколько различных кодов может составить Вася?
- 98) Вася составляет 5-буквенные коды из букв Г, Е, Л, И, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ИЕЙ. Сколько различных кодов может составить Вася?
- 99) Вася составляет 5-буквенные коды из букв Н, И, Ч, Ъ, Я. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ъ и не может содержать сочетания БИЯ. Сколько различных кодов может составить Вася?
- 100) Вася составляет 6-буквенные коды из букв П, А, Н, Е, Л, Ъ. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ъ и не может содержать сочетания ЕАП. Сколько различных кодов может составить Вася?
- 101) Вася составляет 6-буквенные коды из букв Ш, А, Н, Е, Л, Ъ. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ъ и не может содержать сочетания ЕАЬ. Сколько различных кодов может составить Вася?
- 102) Вася составляет 6-буквенные коды из букв Н, И, Г, Р, О, Л. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы О и не может содержать сочетания ОИГ. Сколько различных кодов может составить Вася?
- 103) Вася составляет 7-буквенные коды из букв К, У, П, Ч, И, Х, А. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ч и не может содержать сочетания ИАУ. Сколько различных кодов может составить Вася?
- 104) Вася составляет 7-буквенные коды из букв Н, А, Д, П, И, С, Ъ. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Ъ и не может содержать сочетания БИА. Сколько различных кодов может составить Вася?
- 105) Вася составляет 7-буквенные коды из букв Н, О, Б, Е, Л, И, Й. Каждую букву нужно использовать ровно 1 раз, при этом код не может начинаться с буквы Й и не может содержать сочетания ИЙО. Сколько различных кодов может составить Вася?
- 106) Петя составляет 4-буквенные слова из букв Н, О, Д, А. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные. Сколько различных кодов может составить Петя?
- 107) Петя составляет 5-буквенные слова из букв К, О, Л, У, Н. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные. Сколько различных кодов может составить Петя?
- 108) Петя составляет 6-буквенные слова из букв К, О, М, Е, Т, А. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные. Сколько различных кодов может составить Петя?
- 109) Петя составляет 7-буквенные слова из букв А, Б, Р, И, К, О, С. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные. Сколько различных кодов может составить Петя?
- 110) Маша составляет 6-буквенные коды из букв Р, У, Л, Ъ, К, А. Каждую букву нужно использовать ровно 1 раз, при этом буква Ъ не может стоять на первом месте и после гласной. Сколько различных кодов может составить Маша?

- 111) Маша составляет 7-буквенные коды из букв А, Й, С, Б, Е, Р, Г. Каждую букву нужно использовать ровно 1 раз, при этом буква Й не может стоять на первом месте и перед гласной. Сколько различных кодов может составить Маша?
- 112) Маша составляет 7-буквенные коды из букв В, Е, Н, Т, И, Л, Ь. Каждую букву нужно использовать ровно 1 раз, при этом буква Ь не может стоять на последнем месте и между гласными. Сколько различных кодов может составить Маша?
- 113) Маша составляет 7-буквенные коды из букв П, Е, С, К, А, Р, Ь. Каждую букву нужно использовать ровно 1 раз, при этом буква Ь не может стоять на первом месте, а также перед буквами Е, А и Р. Сколько различных кодов может составить Маша?
- 114) Петя составляет четырёхбуквенные слова перестановкой букв слова АБАК. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?
- 115) **(А. Богданов)** Петя составляет пятибуквенные слова перестановкой букв слова МАРТА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?
- 116) Петя составляет шестибуквенные слова перестановкой букв слова АДЖИКА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?
- 117) 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?
- 118) Петя составляет шестибуквенные слова перестановкой букв слова АВРОРА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?
- 119) **(А.М. Кабанов)** Алексей составляет 5-буквенные слова из букв М, А, Г, И, С, Т, Р. Каждую букву можно использовать не более одного раза, при этом в слове нельзя использовать более одной гласной. Сколько различных кодов может составить Алексей?
- 120) **(А.М. Кабанов)** Юрий составляет 4-буквенные слова из букв П, Р, И, К, А, Э. Каждую букву можно использовать не более одного раза, при этом в слове нельзя использовать более одной гласной. Сколько различных кодов может составить Юрий?
- 121) **(А.Н. Носкин)** Петя составляет шестибуквенные слова перестановкой букв слова ТАРТАР. Сколько всего различных слов может составить Петя?
- 122) **(А.Н. Носкин)** Петя составляет шестибуквенные слова перестановкой букв слова МОЛОКО. Сколько всего различных слов может составить Петя?
- 123) **(А.Н. Носкин)** Петя составляет семибуквенные слова перестановкой букв слова АССАСИН. Сколько всего различных слов может составить Петя?
- 124) **(А.Н. Носкин)** Петя составляет шестибуквенные слова перестановкой букв слова ЧИУАУА. Сколько всего различных слов может составить Петя?
- 125) **(А.Н. Носкин)** Петя составляет семибуквенные слова перестановкой букв слова ТРАТАТА. Сколько всего различных слов может составить Петя?
- 126) **(А.Н. Носкин)** Петя составляет список из 5-буквенных слов, в состав которых входят только буквы А, О, У. Петя расположил слова в **обратном** алфавитном порядке. Вот начало списка:

1. УУУУУ
2. УУУУО
3. УУУУА
4. УУУОУ

.....

Запишите слово, которое стоит в этом списке под номером 100.

- 127) **(А.Н. Носкин)** Петя составляет список из 4-буквенных слов, в состав которых входят только буквы О, С, Е, Н, Ь. Петя расположил слова в **обратном** алфавитном порядке. Вот начало списка:

1. ЪЪЪЪ
2. ЪЪЪС

3. ЪЬЬО

4. ЪЬЬН

5. ЪЬЬЕ

6. ЪЬСЬ

.....

Запишите слово, которое стоит в этом списке под номером 100.

- 128) Артур составляет 5-буквенные коды из букв А, П, О, Р, Т. Каждую букву нужно использовать ровно один раз, при этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 129) Артур составляет 5-буквенные коды перестановкой букв слова ВОРОН. При этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 130) 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?
- 131) Артур составляет 5-буквенные коды перестановкой букв слова АРЕАЛ. При этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 132) Артур составляет 6-буквенные коды перестановкой букв слова АСПЕКТ. При этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 133) Артур составляет 6-буквенные коды из букв З, Д, А, Н, И, Е. Каждую букву нужно использовать ровно один раз, при этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 134) Артур составляет 6-буквенные коды перестановкой букв слова ВОРОТА. При этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 135) Артур составляет 6-буквенные коды перестановкой букв слова КАБАЛА. При этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?
- 136) Василий составляет 4-буквенные коды из букв Г, Е, Р, О, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 137) Василий составляет 4-буквенные коды из букв Г, А, Ф, Н, И, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 138) Василий составляет 4-буквенные коды из букв М, О, И, С, Е, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 139) Василий составляет 4-буквенные коды из букв Е, Н, И, С, Е, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 140) Василий составляет 4-буквенные коды из букв А, Р, С, Е, Н, И, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 141) Василий составляет 4-буквенные коды из букв Б, Е, Р, К, Л, И, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 142) Василий составляет 4-буквенные коды из букв В, А, Я, Ю, Щ, И, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?
- 143) Из букв слова Р У С Т А М составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?



- 144) Из букв А З И М У Т составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 145) Из букв слова Р А Д У Г А составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 146) Из букв слова Р А З М А Х составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 147) Из букв слова К О Р Т И К составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 148) Из букв слова К А Р К А С составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 149) Из букв слова К А Н К А Н составляются 6-буквенные последовательности. Сколько можно составить различных последовательностей, если известно, что в каждой из них содержится не менее 3 согласных?
- 150) **(Б.С. Михлин)** Разведчик кодирует символы текста четырьмя стрелками. Каждая стрелка может иметь четыре положения (направления):  $\uparrow \rightarrow \downarrow \leftarrow$ . Для первой стрелки запрещено положение вверх:  $\uparrow$ . Вторая и третья стрелки не могут находиться в одинаковом положении (направлении). Сколько всего различных символов текста может закодировать разведчик?
- 151) **(Б.С. Михлин)** Разведчик кодирует символы текста четырьмя стрелками. Каждая стрелка может иметь четыре положения (направления):  $\uparrow \rightarrow \downarrow \leftarrow$ . Для первой стрелки запрещено положение вверх:  $\uparrow$ . Стрелки, расположенные через одну, не могут находиться в одинаковом положении (направлении): первая и третья, вторая и четвертая. Сколько всего различных символов текста может закодировать разведчик?
- 152) **(Б.С. Михлин)** Разведчик кодирует символы текста пятью стрелками. Каждая стрелка может иметь четыре положения (направления):  $\uparrow \rightarrow \downarrow \leftarrow$ . Для первой стрелки запрещено положение вверх:  $\uparrow$ . Стрелки, расположенные через одну, не могут находиться в одинаковом положении (направлении): первая и третья, вторая и четвертая, третья и пятая. Сколько всего различных символов текста может закодировать разведчик?
- 153) **(Б.С. Михлин)** Разведчик кодирует символы текста пятью стрелками. Каждая стрелка может иметь четыре положения (направления):  $\uparrow \rightarrow \downarrow \leftarrow$ . Для первой стрелки запрещено положение вверх:  $\uparrow$ . Некоторые стрелки не могут находиться в одинаковом положении (направлении): первая и пятая, вторая и четвертая. Сколько всего различных символов текста может закодировать разведчик?
- 154) **(Б.С. Михлин)** Разведчик кодирует символы текста пятью стрелками. Каждая стрелка может иметь четыре положения (направления):  $\uparrow \rightarrow \downarrow \leftarrow$ . Для первой стрелки запрещено положение вверх:  $\uparrow$ . Запрещено использовать коды, которые являются палиндромами (т.е. одинаково читаются как слева направо, так и справа налево). Сколько всего различных символов текста может закодировать разведчик?
- 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?

- 156) Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 5 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 157) Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 6 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 158) 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.
- 159) Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 8 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 160) Сколько существует чисел, восьмеричная запись которых содержит 5 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 161) Сколько существует чисел, восьмеричная запись которых содержит 6 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 162) Сколько существует чисел, восьмеричная запись которых содержит 7 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 163) Сколько существует чисел, восьмеричная запись которых содержит 8 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 164) Сколько существует чисел, шестнадцатеричная запись которых содержит 3 цифры, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 165) Сколько существует чисел, шестнадцатеричная запись которых содержит 4 цифры, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 166) Сколько существует чисел, шестнадцатеричная запись которых содержит 5 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.
- 167) (**С.А. Скопинцева**) Ваня составляет четырехбуквенные слова из букв О, Б, Ъ, Е, М, причём в каждом слове буква О встречается ровно один раз, а буква Ъ не может стоять на первом месте и не может стоять на последнем месте. Все остальные буквы, могут 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 Ваня?
- 168) Сергей составляет 6-буквенные коды из букв К, Л, Е, Й. Буква Й 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?
- 169) Сергей составляет 5-буквенные коды из букв С, Е, Р, Г, Е, Й. Буква Й 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?
- 170) Сергей составляет 5-буквенные коды из букв Ж, А, Л, Е, Й. Буква Й 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?
- 171) Сергей составляет 5-буквенные коды из букв В, О, Р, О, Б, Е, Й. Буква Й 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?



- 172) Сергей составляет 6-буквенные коды из букв С, О, Л, О, В, Е, Й. Буква Й 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?
- 173) Сергей составляет 6-буквенные коды из букв Е, Л, Е, Й. Буква Й 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?
- 174) Сергей составляет 6-буквенные коды из букв К, А, Л, И, Й. Буква Й может использоваться в коде не более одного раза, при этом она не может стоять на первом месте, на последнем месте и рядом с буквой И. Все остальные буквы могут встречаться произвольное количество раз или не встречаться совсем. Сколько различных кодов может составить Сергей?
- 175) (Е. Джобс) Вася составляет 4-буквенные слова из букв И, Н, С, Т, А, В, К и упорядочивает их по алфавиту. При этом на первом месте может быть только согласная, на последнем - гласная. Вот начало списка:

1. ВААА
2. ВААИ
3. ВАВА

....

Укажите номер слова НИКА в этом списке.

- 176) (Е. Джобс) Вася составляет 4-буквенные слова, в которых есть только буквы С, Ч, И, Т, А, Й, причём буква А может встретиться в каждом слове не более 1 раза. 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 различных слов может написать Вася?
- 177) (Е. Джобс) Настя составляет 6-буквенные слова, в которых есть только буквы Д, Ж, О, Б, С, причём буквы Д, О, С встречаются ровно по одному разу. Буква Ж встречается не более 2 раз, а буква Б может встречаться любое количество раз или не встречаться вовсе. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько различных слов может составить Настя?
- 178) (Е. Джобс) Стасик выписывает все пятисимвольные комбинации, составленные из букв Ш, К, О, Л, А. При этом упорядочивая их по алфавиту. Вот начало списка:

1. ААААА
2. ААААК
3. ААААЛ
4. ААААО
5. ААААШ

...

Определите, сколько слов хотя бы с одной гласной напишет Стасик.

- 179) (Е. Джобс) Стасик выписывает все шестисимвольные комбинации, составленные из букв А, Б, Г, О, Щ. При этом упорядочивая их в обратном алфавитном порядке. Вот начало списка:

1. ШЩЩЩЩЩ
2. ШЩЩЩЩО
3. ШЩЩЩЩГ
4. ШЩЩЩЩБ
5. ШЩЩЩЩА
6. ШЩЩЩЩЩ

...

Определите номер слова ОБЩАГА в этом списке.

180) (**Е. Джобс**) Ипполит составляет 6-буквенные слова, в которых есть только буквы М, Е, Ч, Т, А, причём буква А используется в каждом слове хотя бы 3 раза. 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 различных слов может написать Ипполит?

181) (**Е. Джобс**) Женя составляет слова переставляя буквы З, А, П, И, С, Ъ. Сколько слов может составить Женя, если известно, что Ъ не может стоять на первом месте и после гласной?

182) (**Е. Джобс**) Все 4-буквенные слова, составленные из букв П, Р, В, Д, А, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААА

2. АААВ

3. АААД

4. АААП

5. АААР

6. ААВА

...

Найдите номер первого слова в этом списке, которое не содержит гласных и одинаковых букв.

183) (**Е. Джобс**) Сколько шестнадцатеричных кодов чисел длиной 15 можно составить, если известно, что цифры идут в порядке убывания, при этом четные и нечетные цифры чередуются?

184) Сколько шестнадцатеричных кодов чисел длиной 12 можно составить, если известно, что цифры идут в порядке убывания, при этом четные и нечетные цифры чередуются?

185) Сколько чисел длиной 6 можно составить, если известно, что цифры идут в порядке убывания, при этом четные и нечетные цифры чередуются?

186) Миша составляет 6-буквенные коды из букв Б, А, Л, О, Н. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

187) Миша составляет 6-буквенные коды из букв Б, А, Н, К, И, Р. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

188) Миша составляет 5-буквенные коды из букв С, А, К, У, Р, А. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

189) Миша составляет 5-буквенные коды из букв К, О, Р, Н, Е, Т. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

190) Миша составляет 5-буквенные коды из букв К, А, Л, Ъ, К, А. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

191) Миша составляет 6-буквенные коды из букв С, А, Л, Ъ, С, А. Каждая допустимая гласная буква может входить в код не более одного раза. Сколько кодов может составить Миша?

192) (**А. Богданов**) Марина собирает восьмибуквенные слова из букв своего имени. Первые четыре буквы новых слов берутся из первых четырех букв имени, так чтобы ни одна буква не повторялась. А последние четыре буквы из последних трех букв имени, и они могут многократно повторяться. На каком месте окажется имя МАРИАННА в отсортированном по алфавиту списке сгенерированных слов? Нумерация начинается с 1.

193) (**Е. Джобс**) Сколько существует четных пятиричных чисел длиной 6, начинающихся с цифры 3?

194) (**С. Скопинцева**) Лида составляет слова из букв Л, И, Д, А. Каждая гласная буква встречается в слове не более двух раз. Каждая согласная может стоять в слове на первой позиции, либо не встречаться вовсе. Сколько слов длиной более двух символов может составить Лида?

195) Лида составляет слова из букв С, Е, П, И, Я. Каждая гласная буква встречается в слове не более двух раз. Каждая согласная может стоять в слове на первой позиции, либо не встречаться вовсе. Сколько слов длиной более двух символов может составить Лида?

- 196) Лида составляет слова из букв К, Р, Ы, Ш, А. Каждая гласная буква встречается в слове не более двух раз. Каждая согласная может стоять в слове на первой позиции, либо не встречаться вовсе. Сколько слов длиной более двух символов может составить Лида?
- 197) Ксения составляет слова из букв К, С, Е, Н, И, Я. Каждая гласная буква встречается в слове не более двух раз. Каждая согласная может стоять в слове на первой позиции, либо не встречаться вовсе. Сколько слов длиной более двух символов может составить Ксения?
- 198) **(А. Куканова)** Катя составляет трёхбуквенные слова из букв А, Б, В, Г, Д, причём буквы могут повторяться, но следуют друг за другом в алфавитном порядке. Сколько различных слов может составить Катя?
- 199) **(А. Куканова)** Маша составляет четырёхбуквенные слова из букв А, В, С, D, Е, причём сначала в слове должны быть расположены гласные в алфавитном порядке, затем согласные в обратном алфавитном порядке. Буквы могут повторяться. Слово может состоять только из гласных или только из согласных. Пример подходящего слова: AAEDC. Сколько различных слов может составить Маша?
- 200) **(А. Куканова)** Аня составляет трёхзначные числа в десятичной системе счисления, в которых цифры расположены в порядке неубывания. Сколько различных чисел может составить Аня?
- 201) **(А. Куканова)** Варя составляет пятизначные числа в шестнадцатичной системе счисления, в которых цифры расположены в порядке неубывания. Сколько различных чисел может составить Варя?
- 202) **(А. Куканова)** Маша составляет 4-буквенные слова из букв П, И, Т, О, Н, причём никакие две гласные или две согласные не должны стоять рядом. Каждая буква может использоваться несколько раз или не использоваться вообще. Сколько слов может составить Маша?
- 203) **(А. Куканова)** Лера составляет 5-буквенные слова из букв Л, О, Г, А, Р, И, Ф, М, причём никакие две гласные или две согласные не должны стоять рядом. Буквы в слове не должны повторяться. Сколько слов может составить Лера?
- 204) **(А. Куканова)** Аня составляет 6-значные числа в 10-ичной системе счисления. Цифры в числе не должны повторяться, и никакие две четные или две нечетные цифры не должны стоять рядом. Сколько чисел может составить Аня?
- 205) **(А. Куканова)** Василиса составляет 5-значные числа в 6-ичной системе счисления. Цифры в числе могут повторяться, но никакие две четные или две нечетные цифры не должны стоять рядом. Сколько чисел может составить Василиса?
- 206) **(А. Куканова)** Лена составляет 5-буквенные слова из букв Я, С, Н, О, В, И, Д, Е, Ц, причём слово должно начинаться с согласной и заканчиваться гласной. Первая и последняя буквы слова встречаются в нем только один раз; остальные буквы могут повторяться. Сколько слов может составить Лена?
- 207) **(А. Куканова)** Полина составляет 6-буквенные слова из букв Р, Е, Ж, И, М, Д, Н, О, причём слово должно начинаться с согласной, после которой идёт гласная, и заканчиваться на гласную. Буквы в слове не повторяются. Сколько таких слов может составить Полина?
- 208) **(А. Куканова)** Ада составляет 6-буквенные слова из букв Д, Е, Й, К, С, Т, Р, А. Буква Й встречается в слове ровно один раз, и после неё обязательно идёт согласная. Буквы в слове не повторяются. Сколько слов может составить Ада?
- 209) **(А. Куканова)** Ася составляет 7-буквенные слова из букв А, П, Е, Л, Ъ, С, И, Н. Все буквы слова различны. Буква Ъ, если встречается, стоит между двумя согласными. Сколько таких слов может составить Ася?
- 210) **(А. Куканова)** Ксюша составляет слова, меняя местами буквы в слове МИМИКРИЯ. Сколько различных слов, включая исходное, может составить Ксюша?

- 211) **(А. Куканова)** Даша составляет слова, меняя местами буквы в слове ТИКОК так, что любые две соседние буквы должны быть различны между собой. Сколько слов, включая исходное, может составить Даша?
- 212) **(А. Куканова)** Вероника составляет слова, меняя местами буквы в слове КЛАБХАУС так, что любые две соседние буквы различны между собой. Сколько слов, включая исходное, может составить Вероника?
- 213) **(А. Куканова)** Настя составляет 6-буквенные слова из букв Т, Ь, Ю, Р, И, Н, Г, причём мягкий знак не может стоять в начале слова и после гласной. Все буквы слова различны. Сколько таких слов может составить Настя?
- 214) **(А. Куканова)** Вика составляет 4-буквенные слова из букв В, А, Й, Ф, У, причём слово не может начинаться с буквы Й и не должно содержать сочетаний ВФ и ФВ. Все буквы в слове различны. Сколько таких слов может составить Вика?
- 215) **(А. Куканова)** Рита составляет 4-буквенные слова из букв П, С, К, А, Л, Ь, причём мягкий знак, если встречается, не может стоять в начале слова, а также рядом с ещё одним мягким знаком. Буквы в слове могут повторяться. Сколько таких слов может составить Рита?
- 216) **(А. Куканова)** Света составляет 6-буквенные слова из букв С, О, Л, Н, Ц, Е, причём буква О встречается в слове не более 2 раз, а буква Ц — ровно 1 раз. Буквы могут повторяться. Сколько таких слов может составить Света?
- 217) **(А. Куканова)** Леся составляет слова, содержащие ровно 3 буквы М, из букв Ч, О, А, Н, И, М, Е. Слово может иметь длину от 4 до 6 букв. Сколько слов может составить Леся?
- 218) **(А. Куканова)** Агата составляет 5-буквенные слова из букв П, И, К, А, Ч, У, причём буква У должна встречаться в слове хотя бы два раза. Остальные буквы могут встречаться любое число раз, в том числе не встречаться вообще. Сколько слов может составить Агата?
- 219) **(А. Куканова)** Марта составляет 6-буквенные слова из букв И, Н, Ф, А, причём буква Ф должна встречаться в слове ровно 2 раза. Остальные буквы могут встречаться любое количество раз или не встречаться вообще. Сколько слов может составить Марта?
- 220) **(А. Куканова)** Лиля составляет 5-буквенные слова из букв С, О, Т, К, А, П, Л, З. Слово не должно заканчиваться на гласную и содержать сочетания ЗЛО. Буквы в слове не повторяются. Сколько слов может составить Лиля?
- 221) **(А. Куканова)** Лиза составляет слова из букв О, Н, И, К, С, причём буква С должна встречаться в этих словах ровно 3 раза, а буква О — ровно 1 раз. Длина слова составляет от 4 до 6 букв. Сколько различных слов может составить Лиза?
- 222) **(А. Куканова)** Маша составляет 6-буквенные слова из букв З, Е, Р, К, А, Л, О, содержащие букву К, но не более 4 раз. Остальные буквы не могут повторяться. Сколько различных слов может составить Маша?
- 223) **(А. Куканова)** Аня составляет слова, переставляя буквы в слове ОДЕКОЛОН, избегая слов, где соседние буквы — одинаковые. Сколько различных слов, включая исходное, может составить Аня?
- 224) **(А. Куканова)** Евгения составляет 4-значные числа в 8-ичной системе счисления. Числа должны начинаться с чётной цифры, и цифры в них располагаются в невозрастающем порядке. Сколько различных чисел может составить Евгения?
- 225) **(А. Куканова)** Полина составляет 5-значные числа в 5-ичной системе счисления, которые содержат не более 3 чётных цифр. Сколько различных чисел может составить Полина?
- 226) **(А. Куканова)** Оля составляет 5-буквенные слова из букв К, У, С, А, Т, Ь, причём слова не должны начинаться на мягкий знак и содержать сочетание СУК. Буквы в слове не должны повторяться. Сколько различных слов может составить Оля?
- 227) **(А. Куканова)** Мила составляет 4-значные числа в 8-ичной системе. Сколько различных чисел, делящихся на 4 без остатка, может составить Мила?

- 228) (**А. Богданов**) МАРИНА из букв своего имени составляет слова перестановкой исходных букв.  
Сколько различных слов может составить МАРИНА, если первая буква не может быть гласной?