

Решение задач ЕГЭ из содержательных разделов «Информация и её кодирование», «Моделирование», «Системы счисления»

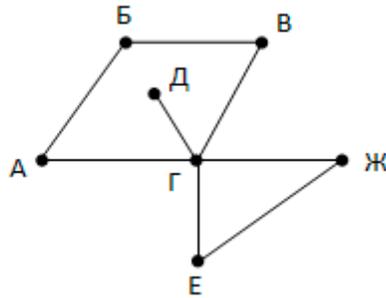
БЕЛЯНЧЕВА СВЕТЛАНА ЮРЬЕВНА,
СТАРШИЙ МЕТОДИСТ ЦИТ
ГАУ ДПО ЯО ИРО

Вариант № 42537

Задание 1

(№ 1594) На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1				9			
п2				16	17		
п3				14			23
п4	9	16	14		30	11	
п5		17		30			
п6				11			15
п7			23			15	

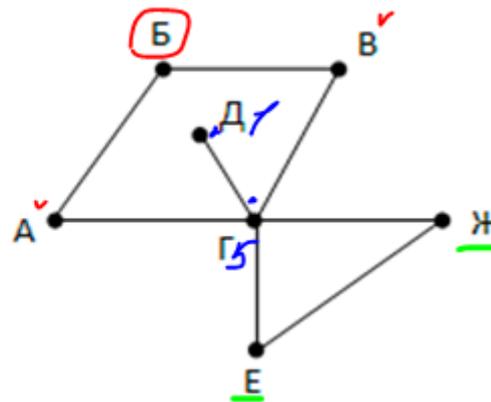


Так как таблицу и схему рисовали

независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину кратчайшего пути из пункта Е в пункт Ж.

	п1	п2	п3	п4	п5	п6	п7
п1				9			
п2				16	17		
п3				14			23
п4	9	16	14		30	11	
п5		17		30			
п6				11			15
п7			23			15	

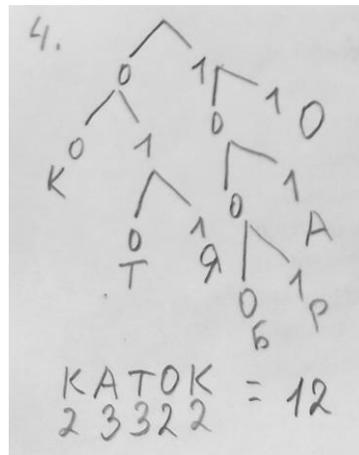
Handwritten annotations: Blue arrows point to p1, p2, p3, p4, p5, p6, p7. Red checkmarks are under p1, p2, p3, p4, p5, p6, p7. A green circle highlights the value 17 in the cell (п5, п2).



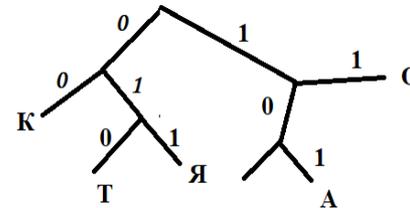
Вариант № 42537

Задание 4

(№ 1682) По каналу связи передаются сообщения, содержащие только семь букв: А, Б, К, О, Т, Р, Я. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: А – 101, О – 11, Я – 011. Какое наименьшее количество двоичных знаков потребуется для кодирования слова КАТОК?



Решение: нарисуем дерево, где отметим коды известных букв, и короткие варианты свободных кодов возьмем для букв К и Т. Таким образом, получим К – 00, Т – 010. Т.к. К встречается в слове два раза, для нее код короче. Для кодирования слова КАТОК понадобится минимум 12 знаков:
001010101100

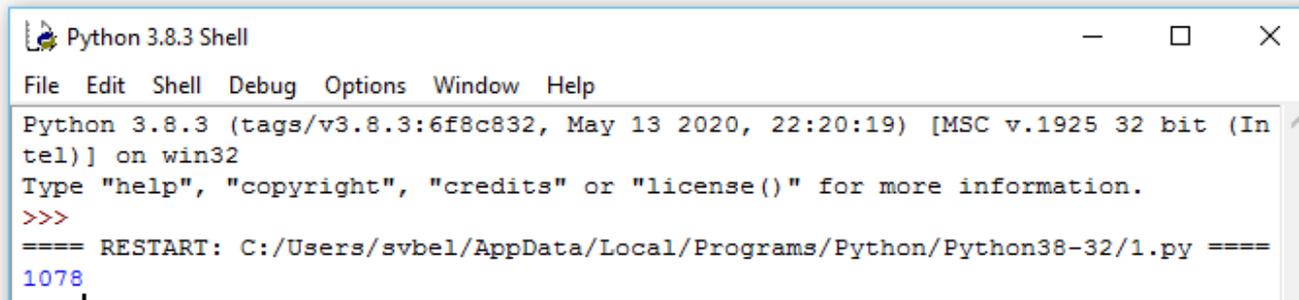


Вариант № 42537

Задание 8

(№ 4130) (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.

```
k=0
for i1 in range(0,4):
    for i2 in range(0,4):
        for i3 in range(0,4):
            for i4 in range(0,4):
                for i5 in range(4,7):
                    for i6 in range(4,7):
                        for i7 in range(4,7):
                            for i8 in range(4,7):
                                if i1!=i2 and i2!=i3 and i3!=i4 and i1!=i3 and i1!=i4 and i4!=i2:
                                    k+=1
                                    if i5==4 and i6==6 and i7==6 and i8==4 and i1==2 and i2==0 and i3==3 and i4==1:
                                        print(k)
```



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/svbel/AppData/Local/Programs/Python/Python38-32/1.py ====
1078
```


Задание 11

(№ 3336) Каждый сотрудник предприятия получает электронный пропуск, на котором записаны личный код, состоящий из двух частей. Первая часть кода содержит 7 символов, каждый из которых может быть одной из 26 заглавных латинских букв. Вторая часть кода содержит 4 символа, каждый из которых может быть одной из десятичных цифр. При этом в базе данных сервера формируется запись, содержащая этот код и дополнительную информацию о пользователе. Для представления кода используют посимвольное кодирование, все символы в пределах одной части кода кодируют одинаковым минимально возможным для этой части количеством битов, а для кода в целом выделяется минимально возможное целое количество байтов. Для хранения данных о 45 пользователях потребовалось 2385 байт. Сколько байтов выделено для хранения дополнительной информации об одном пользователе? В ответе запишите только целое число – количество байтов.

Алфавит 26, значит ищем ближайшую большую степень 2ки, $2^5=32$, $i=5$

На первую часть кода отводится 7 символов, получается $5*7=35$ бит

Алфавит 10, значит ищем ближайшую большую степень 2ки, $2^4=16$, $i=4$

На вторую часть кода отводится 4 символов, получается $4*4=16$ бит

Всего на код $35+16=51$ бит $/8=6,4$ байт= 7 байт

Для хранения данных о 45 пользователях потребовалось 2385 байт, на 1 пользователя необходимо $2385/45=53$ байта

53 байта – 7 байт = 46 байт – доп. информация

Ответ 46 байт

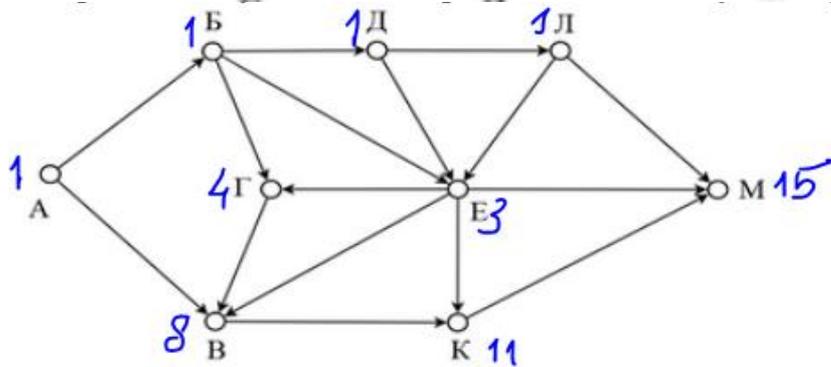
Задание 13 Графы. Поиск количества путей

Даны ориентированные графы. Движение возможно в указанном стрелками направлении. Необходимо подсчитать количество различных путей из пункта А в пункт В.

Считаем пути последовательно от ближайших пунктов.

Количество путей до пункта равно сумме путей до предыдущих пунктов.

(Е. Джобс) На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М?



Решение с помощью программирования.

Строка содержит название вершины из которой мы начали путь и вершины, в которые можно из нее сразу прийти:

`s = 'АБВ БДЕГ ВК ГВ ДЛЕ ЕВКМ КМ ЛЕМ'`

Создается словарь, где у каждого элемента ключ – это обозначение вершины, а значение – это список вершин, куда можно прийти за один шаг из данной вершины.

`D = {'А' : 'БВ', 'Б' : 'ДЕГ', 'В' : 'К', 'Г' : 'В', 'Д' : 'ЛЕ', 'Е' : 'ВКМ', 'К' : 'М', 'Л' : 'ЕМ'} или`

`D = {c[0]:c[1:] for c in s.split()}`

Рекурсивная процедура для подсчета всех возможных путей.

`def f(s,end):`

`if s[-1] == end : return 1`

`return sum(f(s+c,end) for c in D[s[-1]])`

Основная программа:

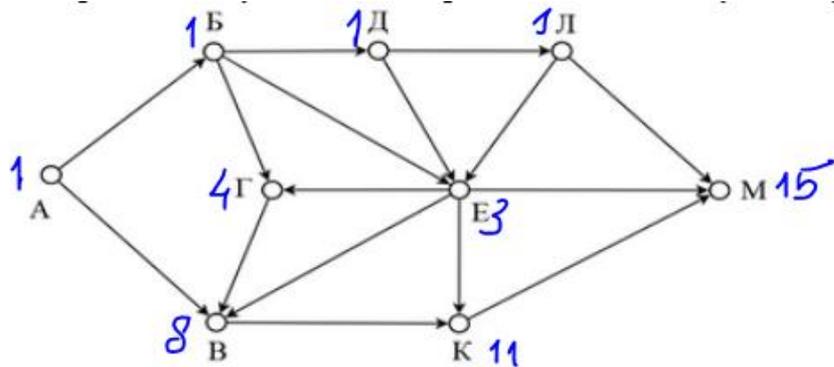
`print(f('А','М'))`

Если последний пункт пути `s[-1]` равен

конечному пункту, то `f = 1`, иначе

Суммируем разные количества путей, которые

можно получить из следующих вершин.



```

s = 'АВВ БДЕГ ВК ГВ ДЛЕ ЕВГКМ КМ ЛЕМ'
d = {c[0]:c[1:]for c in s.split()}
print('d=',d)
def f(s,end):
    if s[-1]==end:
        print(s)
        return 1
    return sum(f(s+c,end) for c in d[s[-1]])
print(f('A','M'))

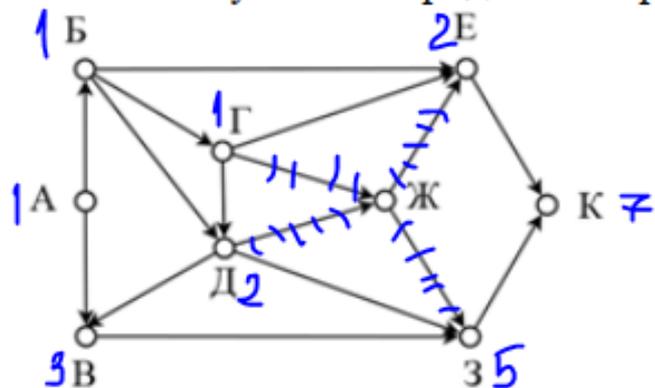
```

```

/13.py -----
d= {'A': 'ВВ', 'Б': 'ДЕГ', 'В': 'К', 'Г': 'В', 'Д': 'ЛЕ',
'E': 'ВГКМ', 'К': 'М', 'Л': 'ЕМ'}
АБДЛЕВКМ
АБДЛЕГВКМ
АБДЛЕКМ
АБДЛЕМ
АБДЛМ
АБДЕВКМ
АБДЕГВКМ
АБДЕКМ
АБДЕМ
АБЕВКМ
АБЕГВКМ
АБЕКМ
АБЕМ
АБГВКМ
АВКМ
15
...

```

(№ 4165) (Е. Джобс) На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К, не проходящих через Ж?



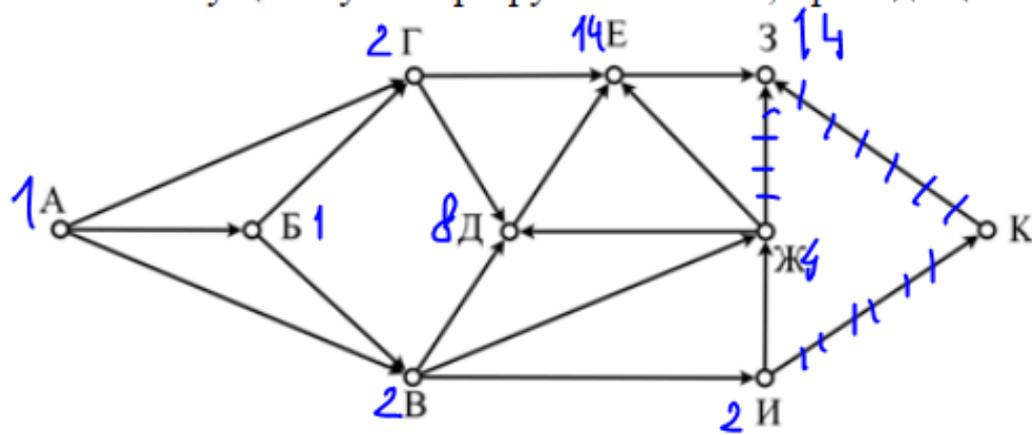
```
s = 'АВВ БДЕГ ВЗ ГЕДЖ ДЖЗВ ЕК ЖЕЗ ЭК'
d = {c[0]:c[1:]for c in s.split()}

def f(s,end):
    if s[-1]==end :
        return 'Ж' not in s
    return sum(f(s+c,end) for c in d[s[-1]])

print(f('А','К'))
```

```
type
>>>
====
7
>>> |
```

(№ 3634) (Е. Джобс) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует маршрутов из А в З, проходящих через город Е?



```
s = 'АБВГ БГВ ВДЖИ ГЕД ДЕ ЕЗ ЖЕЗД ИЖК КЗ'
d = {c[0]:c[1:]for c in s.split()}

def f(s,end):
    if s[-1]==end :
        return 'E' in s
    return sum(f(s+c,end) for c in d[s[-1]])

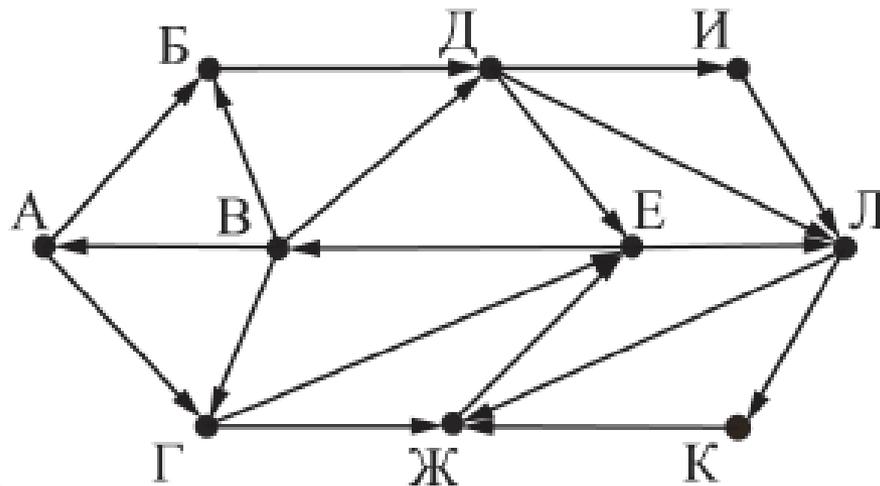
print(f('A','З'))
```

```
>>>
=====
14
>>>
>>> |
```

13

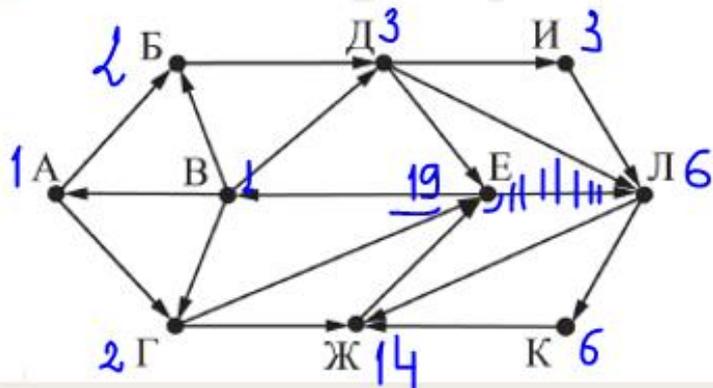
На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Определите количество различных путей ненулевой длины, которые начинаются и заканчиваются в городе Е, не содержат этот город в качестве промежуточного пункта и проходят через промежуточные города не более одного раза.



13 На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

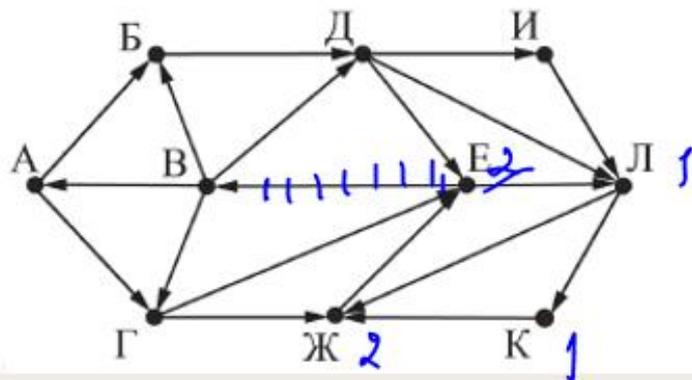
Определите количество различных путей ненулевой длины, которые начинаются и заканчиваются в городе Е, не содержат этот город в качестве промежуточного пункта и проходят через промежуточные города не более одного раза.



$$19 + 2 = 21$$

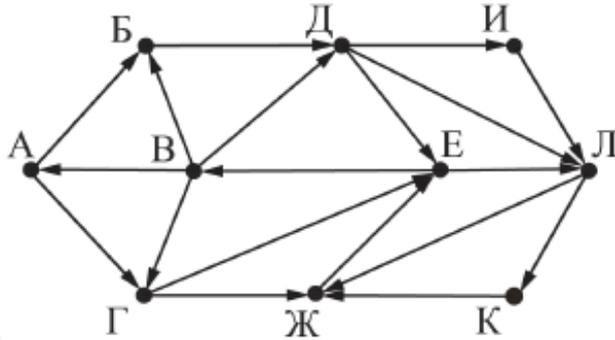
13 На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Определите количество различных путей ненулевой длины, которые начинаются и заканчиваются в городе Е, не содержат этот город в качестве промежуточного пункта и проходят через промежуточные города не более одного раза.



Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

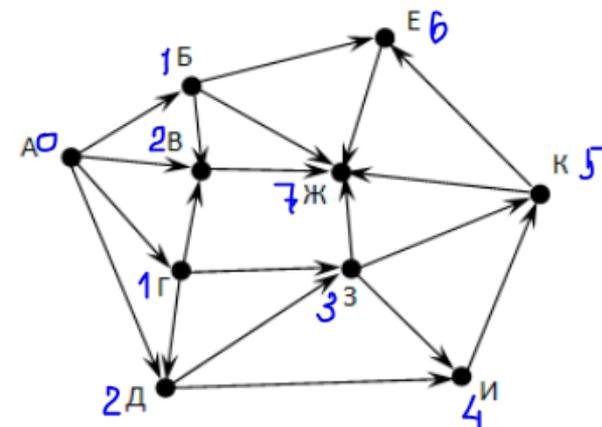
Определите количество различных путей ненулевой длины, которые начинаются и заканчиваются в городе Е, не содержат этот город в качестве промежуточного пункта и проходят через промежуточные города не более одного раза.



```
s = 'АВГ ВД ВАВДГ ГЕЖ ДЕИЛ ЕВЛ ИЛ КЖ ЖЕ ЛЖК'  
d = {c[0]:c[1:] for c in s.split()}  
  
def f(s,end):  
    if s[-1]==end :  
        return 1  
    return sum(f(s+c,end) for c in d[s[-1]] if c not in s)  
#if c not in s проверяет посещение промежуточных пунктов  
#не более 1 раза  
print(f('В','Е')+ f('Л','Е'))
```

```
AMDD  
Type  
>>>  
====  
21  
>>> |
```

(№ 2146) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Какова длина самого длинного пути из города А в город Ж? Длиной пути считать количество дорог, составляющих этот путь.



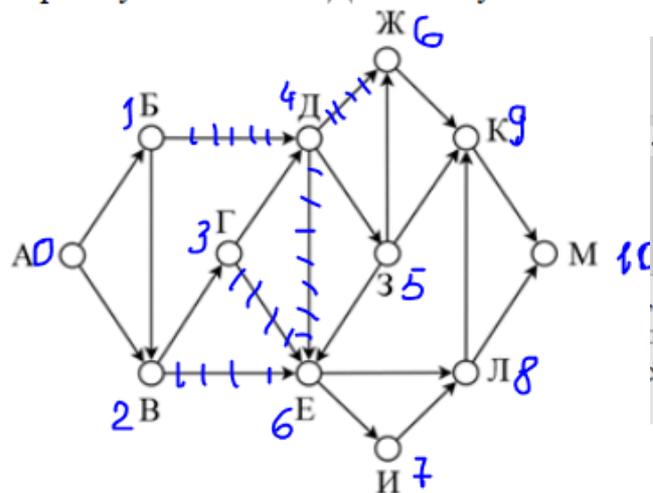
```
s = 'АВВГД БЕЖВ ВЖ ГВЗД ДЗИ ЕЖ ИК КЖЕ ЗЖКИ'
d = {c[0]:c[1:] for c in s.split()}

def f(s,end):
    if s[-1]==end:
        return len(s)-1
    return max(f(s+c,end) for c in d[s[-1]])

print(f('А', 'Ж'))
```

```
tel) |
Type
>>>
=====
7
>>> |
```

(№ 5472) (Е. Джобс) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Какова **длина** самого длинного пути из города А в город М, проходящего одновременно через пункты В и З? Длиной пути считать количество дорог, составляющих этот путь.



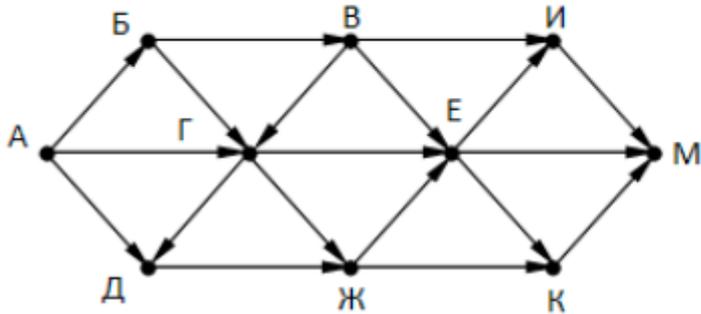
```
s = 'АВВ ВВ ВГ ГД ДЗ ЕЛИ ЗКЕ ИЛ КМ ЛМК ЖК'#нет запрещенных пунктов
d = {c[0]:c[1:] for c in s.split()}

def f(s,end):
    if s[-1]==end:
        return len(s)-1
    return max(f(s+c,end) for c in d[s[-1]])

print(f('А', 'М'))
```

```
IDLE Shell 3.10.0
File Edit Shell Debug
Python 3.10.0
AMD64] on w:
Type "help",
>>>
===== RESTART:
10
|
```

(№ 4424) (Д. Муфаззалов) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, содержащих ровно семь городов, включая города А и М?

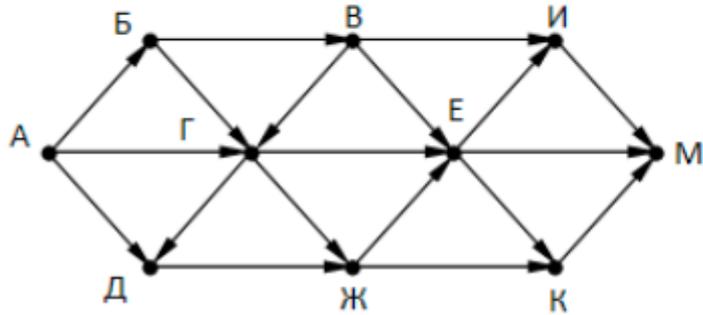


Для каждого пункта будем определять число маршрутов всех возможных длин в этот город из исходного пункта.

Кол-во маршрутов : [длина маршр. 1]^[кол-во маршр. этой длины] х [длина маршр. 2]^[кол-во маршр. этой длины] х [длина маршр. 3]^[кол-во маршр. этой длины] и т.д.

При переходе к следующему пункту **длина маршрута увеличивается на 1, количество сохраняется.**

(№ 4424) (Д. Муфаззалов) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, содержащих ровно семь городов, включая города А и М?



$$А: 1^1$$

$$Б: А = 2^1$$

$$В: Б = 3^1$$

$$Г: А \cdot Б \cdot В = 2^1 \cdot 3^1 \cdot 4^1$$

$$Д: А \cdot Г = 2^1 \cdot 3^1 \cdot 4^1 \cdot 5^1$$

$$Ж: Г \cdot Д = 3^1 \cdot 4^1 \cdot 5^1 \cdot 3^1 \cdot 4^1 \cdot 5^1 \cdot 6^1 = 3^2 \cdot 4^2 \cdot 5^2 \cdot 6^1$$

$$Е: В \cdot Г \cdot Ж = 4^1 \cdot 3^1 \cdot 4^1 \cdot 5^1 \cdot 4^2 \cdot 5^2 \cdot 6^2 \cdot 7^1 = 3^1 \cdot 4^4 \cdot 5^3 \cdot 6^2 \cdot 7^1$$

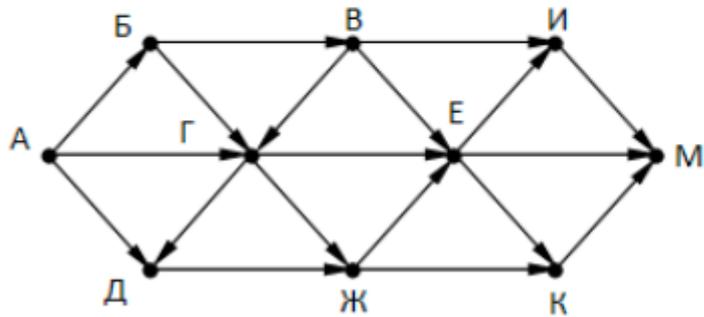
$$И: В \cdot Е = 4^1 \cdot 4^1 \cdot 5^4 \cdot 6^3 \cdot 7^2 \cdot 8^1 = 4^2 \cdot 5^4 \cdot 6^3 \cdot 7^2 \cdot 8^1$$

$$К: Е \cdot Ж = 4^1 \cdot 5^4 \cdot 6^3 \cdot 7^2 \cdot 8^1 \cdot 4^2 \cdot 5^2 \cdot 6^2 \cdot 7^1 = 4^3 \cdot 5^6 \cdot 6^5 \cdot 7^3 \cdot 8^1$$

$$М: И \cdot Е \cdot К = 5^2 \cdot 6^4 \cdot 7^3 \cdot 8^2 \cdot 9^1 \cdot 4^1 \cdot 5^4 \cdot 6^3 \cdot 7^2 \cdot 8^1 \cdot 5^3 \cdot 6^6 \cdot 7^5 \cdot 8^3 \cdot 9^1 = 4^1 \cdot 5^9 \cdot 6^{13} \cdot 7^{10} \cdot 8^6 \cdot 9^2$$

Из города А в город М есть 10 маршрутов, проходящих через 7 городов.

(№ 4424) (Д. Муфаззалов) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, содержащих ровно семь городов, включая города А и М?



```
s = 'АБДГ БВГ ВИЕГ ГДЕЖ ДЖ ЕИКМ ИМ КМ ЖЕК'
d = {c[0]:c[1:] for c in s.split()}

def f(s,end):
    if s[-1]==end:
        return len(s)==7
    return sum(f(s+c,end) for c in d[s[-1]])

print(f('А', 'М'))
```

```
>>>
====
10
>>> |
Type
```

Содержательный раздел «Системы счисления»

Кодификатор ЕГЭ 2023 г. ИНФОРМАТИКА, 11 класс

Раздел 2. Перечень элементов содержания

1.4 Системы счисления

1.4.1 Позиционные системы счисления

1.4.2 Двоичное представление информации

Спецификация КИМ ЕГЭ 2023 г.

Обобщённый план варианта КИМ ЕГЭ 2023 года по ИНФОРМАТИКЕ

№	Проверяемые элементы содержания	Коды эл-тов содер-я	Коды уров. подгот.	Уровень слож.	Треб-ся ПО	Макс. балл	Время (мин.)
14	Знание позиционных систем счисления	1.4.1	1.1.3	П	нет	1	3

Что нужно знать:

Позиционные системы.

Развернутая запись числа – разложение по разрядам.

Недесятичные системы счисления.

Перевод из любой системы счисления в десятичную. Функция `int()`. Ограничение для `int()`: $2 \leq CC \leq 36$.

Разбиение на цифры – деление на основание `CC` с остатком.

Обратите внимание, что данная программа перевода работает только для $CC < 10$:

Если вместо строки использовать список, то получим универсальную программу перевода

Функции перевода: `bin()`, `oct()`, `hex()`. В результате получим строку с префиксом, указывающим на новую `CC`. Поэтому функции используются со срезом `[2:]`

```
x = 25
s = ''
while x > 0 :
    s += str(x%5)
    x //= 5
print(s[::-1])
```

```
x = 188
s = []
while x > 0 :
    s += [x%16]
    x //= 16
print(s[::-1])
```

```
>>> bin(200)[2:]
'11001000'
>>> oct(200)[2:]
'310'
>>> hex(200)[2:]
'c8'
```

Что нужно знать:

Определение окончания числа, длины числа.

В десятичной СС:

$x\%10$ – последняя цифра

$x//10\%10$ – вторая с конца

$x//100\%10$ – третья с конца

В общем случае:

$x\%N$ – последняя цифра

$x//N\%N$ – вторая с конца

$x//N^2\%N$ – третья с конца

Длина числа – количество цифр в числе.

Общие формы для записи чисел:

$$a^N = 10 \underbrace{\dots 0}_N a$$

$$a^N - a^M = \underbrace{(a-1) \dots (a-1)}_{N-M} \underbrace{0 \dots 0}_M a$$

14 Значение арифметического выражения: $49^7 + 7^{21} - 7$ – записали в системе счисления с основанием 7. Сколько цифр 6 содержится в этой записи?

1. Переведем все числа к степеням семерки:

$$(7^2)^7 + 7^{21} - 7 = 7^{14} + 7^{21} - 7$$

2. Расставим степени в порядке убывания:

$$7^{21} + 7^{14} - 7$$

3. Выполним вычитание:

$$7^{14} - 7 = 666666666666660$$

Ответ: 13

```
x = 49**7+1**21-7
s = []
while x > 0 :
    s += [x%7]
    x //= 7
print(s.count(6))
```

Type "r
>>>
==== RE
13
>>> |

$8^{500} + 4^{1500} - 2^{100}$ Сколько «1» в двоичной записи числа?

1. Приведем все числа к степеням «2»:

$$8^{500} + 4^{1500} - 2^{100} = 2^{1500} + 2^{3000} - 2^{100}$$

2. Расставим степени в порядке убывания:

$$2^{3000} + 2^{1500} - 2^{100}$$

3. Рассмотрим разность:

Правило 2: количество «0» = 100; количество «1» = 1500 - 100 = 1400

4. Рассмотрим сумму:

$$\underbrace{10..0}_{3000} + \underbrace{1..10..0}_{1400} \underbrace{0..0}_{100} = 1 \underbrace{0..0}_{?} \underbrace{1..1}_{1400} \underbrace{0..0}_{100}$$

Количество «0» над знаком «?»: $3000 - 1400 - 100 = 1500$

5. Посчитаем все цифры:

«0»: $1500 + 100 = 1600$

«1»: $1 + 1400 = 1401$

```
x = 8**500 + 4**1500 - 2**100
print(bin(x)[2:].count('0'))
print(bin(x)[2:].count('1'))
```

```
====
1600
1401
>>> |
```

Позиционные системы счисления с любыми основаниями

(№ 5321) (Е. Джобс) Значение выражения $7 \cdot 5^{1984} - 6 \cdot 25^{777} + 5 \cdot 125^{333} - 4$ записали в системе счисления с основанием 5. Найдите сумму разрядов полученной записи. Ответ представьте в десятичной системе счисления.

```
x = 7*5**1984-6*25**777+5*125**333-4
sm = 0
while x>0 :
    sm = sm + x%5
    x = x // 5
print(sm)
```

```
>>>
==== R:
5718
>>> |
```

(№ 3862) (А. Кабанов) Значение выражения $64^{11} - 4^{10} + 96 - x$ записали в четверичной системе счисления, при этом сумма цифр в записи оказалась равной 71. При каком минимальном натуральном x это возможно?

```
for i in range(1,100):
    x = 64**11-4**10+96-i
    a = []
    while x > 0 :
        a = [x%4] + a
        x = x // 4
    if sum(a) == 71 :
        print(i)
```

```
====
16
28
31
64
76
79
88
91
94
```

(№ 3670) (П.М. Волгин) Значение арифметического выражения $17^5 + 85^8 - 10$ записали в системе счисления с основанием 17. В этой записи помимо цифр от 0 до 9 могут встречаться цифры из списка: A, B, C, D, E, F, G, которые имеют числовые значения от 10 до 16 соответственно. Сколько цифр G встречается в этой записи?

```
x = 17**5+85**8-10
a = []
while x > 0 :
    a = [x%17] + a
    x = x // 17
print(a.count(16))
```

```
>>>
Type:
====:
5
>>> |
```

(№ 3668) Число 1988 записали в системах счисления с основаниями от 2 до 10 включительно. При каких основаниях в записи этого числа нет двух одинаковых цифр, стоящих рядом? В ответе укажите сумму всех подходящих оснований.

```
t = 1988
sm = 0
for n in range(2,10) :
    x = t
    a = []
    # flag = True
    while x >0 :
        a = [x%n]+a
        x //= n
    print(a)
    if all(a[i] != a[i+1] for i in range(len(a)-1)) :
        sm += n
## for i in range(len(a)-1) :
##     if a[i] == a[i+1] : flag = False
##     if flag == True : sm += n
print(sm)
```

```
>>>
AMD64)] on win32
Type "help", "copyright", "credits
==== RESTART: C:/Users/student/App:
[1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0]
[2, 2, 0, 1, 1, 2, 2]
[1, 3, 3, 0, 1, 0]
[3, 0, 4, 2, 3]
[1, 3, 1, 1, 2]
[5, 5, 4, 0]
[3, 7, 0, 4]
[2, 6, 4, 8]
22
>>> |
```

Уравнения с данными в различных системах счисления

(№ 3576) Решите уравнение $224_x + 1 = 101_8$. Ответ запишите в десятичной системе счисления.

```
#переводим числа в 10-ую СС
x3 = int('101',8)
# число 224 не может быть записано в 2,3,4 -ых СС
for x in range (5,20):
    x1 = int('224',x)
    if x1 + 1 == x3 :
        print(x)
```

>>> ==
5
>>> |

```
#переводим числа в 10-ую СС, используя развернутую запись
x3 = 1*8**2 + 1*8**0
# число 224 не может быть записано в 2,3,4 -ых СС
for x in range (5,20):
    x1 = 2*x**2 + 2*x**1 + 4*x**0
    if x1 + 1 == x3 :
        print(x)
```

>>> ==
5
>>> |

Неизвестные цифры в записи чисел в разных системах

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

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

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

Решение.

1) запишем оба слагаемых в развернутой записи в системе счисления с основанием 15:

$$123x5_{15} + 1x233_{15} = (1 \cdot 15^4 + 2 \cdot 15^3 + 3 \cdot 15^2 + x \cdot 15 + 5) + (1 \cdot 15^4 + x \cdot 15^3 + 2 \cdot 15^2 + 3 \cdot 15 + 3) =$$

(все без x) + (все с x)

$$= (2 \cdot 15^4 + 2 \cdot 15^3 + 5 \cdot 15^2 + 3 \cdot 15 + 8) + (x \cdot 15^3 + x \cdot 15) = (101250 + 6750 + 1125 + 45 + 8) + x \cdot (3375 + 15) = 109178 + 3390 \cdot x$$

2) нам нужно, чтобы выражение $109178 + 3390 \cdot x$ делилось на 14

3) остаток от деления 109178 на 14 равен 6; остаток от деления 3390 на 14 равен 2

для того чтобы выражение делилось на 14, остаток от его деления на 14 должен быть равен 0 (14, 28 и т.д.)

Попробуем сложить остатки.

$6 + 2x = 0$, даст нам отрицательное значение x , значит нужно взять следующее значение остатка

$$6 + 2x = 14 \Leftrightarrow 2x = 8 \Leftrightarrow x = 4.$$

$$109178 + 3390 \cdot 4 = 122738. \quad 122738 : 14 = 8767$$

Ответ: 8767

Неизвестные цифры в записи чисел в разных системах

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

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

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

```
>>>
for x in '0123456789abcde' :
    a = int(f'123{x}5',15) + int(f'1{x}233',15)
    # a = int('123'+x+'5', 15) + int('1'+x+'233', 15)
    if a%14 == 0 :
        print(a//14)
>>>
===== RES'
8767
10462
>>> |
```

```
>>>
for x in range (15):
    a = 1*15**4 + 2*15**3 + 3*15**2 + x*15 + 5
    b = 1*15**4 + x*15**3 + 2*15**2 + 3*15 + 3
    if (a+b) %14 == 0:
        print( (a+b) // 14 )
        break
>>>
===== RE
8767
>>> |
```

(№ 5496) (В. Шубинкин) Операнды арифметического выражения записаны в системах счисления с основаниями 25 и 11.

$$7y23x5_{25} + 67x9y_{11}$$

В записи чисел переменными x и y обозначены допустимые в данных системах счисления неизвестные цифры. Определите значения x , y , при которых значение данного арифметического выражения кратно 131. Для найденных значений x , y вычислите частное от деления значения арифметического выражения на 131 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

```
for x in range(25):
    for y in range(11):
        a = 7*25**5 + y*25**4 + 2*25**3 + 3*25**2 + x*25 + 5
        b = 6*11**4 + 7*11**3 + x*11**2 + 9*11 + y
        if (a+b) % 131 == 0:
            print( (a+b) // 131 )
            break
```

>>>

==== RESTART:

552647

531785

>>>

(№ 5497) (В. Шубинкин) Числа М и N записаны в системах счисления с основаниями 15 и 13 соответственно.

$$M = 2y23x5_{15}, N = 67x9y_{13}$$

В записи чисел переменными x и y обозначены допустимые в данных системах счисления неизвестные цифры. Определите наименьшее значение натурального числа A, при котором существуют такие x, y, что $M + A$ кратно N.

```
for a in range(10000):
    for x in range(15):
        for y in range(13):
            m = 2*15**5 + y*15**4 + 2*15**3 + 3*15**2 + x*15 + 5
            n = 6*13**4 + 7*13**3 + x*13**2 + 9*13 + y
            if (m+a) % n == 0:
                print(a)
                break
```

```
==== RESTAF
1535
3210
3730
4885
5236
6560
6742
8235
8248
9754
9910
>>> |
```