

## Задание

Тут должна быть картинка с заданием. Если она не отобразится, то у меня 1 вариант

## Решение

*Задание 1: определить, какую информацию можно получить по криптограмме.*

Для красивого вывода подключаем tabulate. Далее нам еще понадобятся модуль numpy.

```
from tabulate import tabulate
import numpy as np
tfmt = 'github'
ffmt = '.3f'
```

Вводим исходные данные.

*# пространство сообщений*

```
m = ['x', 'y', 'z']
mSize = len(m)
pM = [1/2, 1/4, 1/4]
```

```
print("Пространство сообщений (открытых текстов) M:")
print(tabulate([m], tablefmt='plain'))
```

```
print("Вероятности выбора сообщений:")
print(tabulate([pM], headers=m, tablefmt=tfmt))
```

*# пространство ключей*

```
k = ['k1', 'k2', 'k3']
kSize = len(k)
pK = [1/4, 3/8, 3/8]
```

```
print()
print("Пространство ключей K:")
print(tabulate([k], tablefmt='plain'))
```

```
print("Вероятности выбора ключей:")
print(tabulate([pK], headers=m, tablefmt=tfmt))
```

*# пространство криптограмм*

```
c = ['a', 'b', 'c']
cSize = len(c)
func = [['a', 'b', 'a'],
        ['c', 'b', 'c'],
        ['a', 'a', 'b']]
```

```
print()
```

```
print("Пространство криптограмм C:")
print(tabulate([c], tablefmt='plain'))
```

```
print("Функция шифрования E(k):")
print(tabulate(func, headers=m, showindex=k, tablefmt=tfmt))
```

Пространство сообщений (открытых текстов) M:

```
x y z
Вероятности выбора сообщений:
| x | y | z |
|----|----|----|
| 0.5 | 0.25 | 0.25 |
```

Пространство ключей K:

```
k1 k2 k3
Вероятности выбора ключей:
| x | y | z |
|----|----|----|
| 0.25 | 0.375 | 0.375 |
```

Пространство криптограмм C:

```
a b c
Функция шифрования E(k):
| x | y | z | |
|---|---|---|---|
| k1 | a | b | a |
| k2 | c | b | c |
| k3 | a | a | b |
```

Далее находим нужные вероятности по методичке.

```
pC = [0] * cSize
for i in range(0, kSize):
    for j in range(0, mSize):
        pC[c.index(func[i][j])] += pK[i] * pM[j]
```

```
print()
print("Распределение вероятностей на C:")
print(tabulate([pC], headers=c, tablefmt=tfmt, floatfmt=ffmt))
```

```
pCM = list() # помогите, я не умею нормально задавать пустые матрицы
for i in range(0, mSize):
    pCM.append(list())
    for j in range(0, cSize):
        pCM[i].append(0)

for i in range(0, kSize):
    for j in range(0, mSize):
        pCM[j][c.index(func[i][j])] += pK[i]
```

```

print()
print("Вероятность выбора шифротекста C при известном открытом тексте M:")
print(tabulate(pCM, headers=c, showindex=m, tablefmt=tfmt,
floatfmt=ffmt))

pMC = list()
for i in range(0, cSize):
    pMC.append(list())
    for j in range(0, mSize):
        pMC[i].append(0)
        pMC[i][j] = pM[j] * pCM[j][i] / pC[i]

print()
print("Вероятность открытых текстов M при данных шифротекстах C:")
print(tabulate(pMC, headers=m, showindex=c, tablefmt=tfmt,
floatfmt=ffmt))

```

Распределение вероятностей на C:

	a	b	c
-----	-----	-----	-----
	0.469	0.250	0.281

Вероятность выбора шифротекста C при известном открытом тексте M:

	a	b	c
----	-----	-----	-----
x	0.625	0.000	0.375
y	0.375	0.625	0.000
z	0.250	0.375	0.375

Вероятность открытых текстов M при данных шифротекстах C:

	x	y	z
----	-----	-----	-----
a	0.667	0.200	0.133
b	0.000	0.625	0.375
c	0.667	0.000	0.333

Анализируя полученные результаты, приходим к следующим выводам:

- если мы получаем шифртекст *a*, то более вероятно, что соответствующий открытый текст - *x*, чем *y* или *z*
- по полученному шифртексту *b* можно сказать, что исходное сообщение - точно не *x*, а вероятнее *y*, реже - *z*
- шифртекст *c* позволяет сделать вывод, что сообщение не являлось *y*, а было скорее всего *x*, реже - *z*

*Задание 2: найти энтропии для каждого из пространств.*

Легко находим энтропию по данной в методичке формуле.

```
spacesNames = ['H(M)', 'H(K)', 'H(C)']
spaces = [np.array(s) for s in [pM, pK, pC]]
entropy = [(-1) * np.sum(s * np.log2(s)) for s in spaces]
print(tabulate([entropy], headers=spacesNames, tablefmt=tfmt,
floatfmt=ffmt))
```

H(M)	H(K)	H(C)
1.500	1.561	1.527

Энтропия каждого из пространств примерно равна 1.5. Следовательно, "утекает" примерно 1.5 бита информации о ключе и соответствующем открытом тексте.

*Задание 3: найти неопределенность ключа и сделать вывод.*

Неопределенность ключа - величина  $H(K|C)$ . Находится она по формуле  $H(K|C) = H(M) + H(K) - H(C)$ .

```
print(f"H(K|C) = {entropy[0] + entropy[1] - entropy[2]}")
```

$H(K|C) = 1.5341744978062763$

У меня проблемы с выводом. У вас в методичке  $H(C) > H(K|C)$  и все сходится. А у меня вроде как наоборот получилось.

$H(C) = 1.527$

$H(K|C) = 1.534$

Не понятно - нормально это или нет. И непонятно, что писать.