

Задание 1.

Исходные данные:

Из колоды в 52 карты извлекаются случайным образом 4 карты. а) Найти вероятность того, что все карты – крести. б) Найти вероятность, что среди 4-х карт окажется хотя бы один туз.

Решение:

а) Найти вероятность того, что все карты – крести.

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import math
>>> from math import factorial
>>> m = 52
>>> k = 4
>>> def combinations(m, k):
...     try:
...         m = int(input("Введите число: "))
...         k = int(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     s = int(factorial(m) / (factorial(k) * factorial(m - k)))
...     return s
...
>>> print(combinations(m, k))
Введите число: 52
Введите число: 4
270725
>>>
```

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> n = 52
>>> a = np.random.randint(1, 5, size=n)
>>> a
array([2, 4, 4, 2, 1, 4, 3, 1, 1, 1, 3, 1, 3, 4, 4, 1, 4, 2, 2, 3, 2, 2,
       2, 4, 2, 3, 1, 3, 2, 1, 1, 3, 3, 3, 3, 3, 4, 1, 3, 3, 2, 1, 4, 1,
       1, 2, 2, 3, 4, 3, 2, 3])
>>> m = len(a[a==4])
>>> m
10
>>> w = m / n
>>> w
0.19230769230769232
>>>
```

б) Найти вероятность, что среди 4-х карт окажется хотя бы один туз

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import math
>>> from math import factorial
```

```

>>> m = 4
>>> def perest(m):
...     try:
...         m = int(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     s = int(factorial(m))
...     return s
...
>>> print(perest(m))
Введите число: 4
24
>>>

```

```

Python 3.8.5 (default, May 27 2021, 13:30:53)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> from math import factorial
>>> m = 4
>>> k = 1
>>> def combinations(m, k):
...     try:
...         m = int(input("Введите число: "))
...         k = int(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     s = int(factorial(m) / (factorial(k) * factorial(m - k)))
...     return s
...
>>> print(combinations(m, k))
Введите число: 4
Введите число: 4
1
>>>

```

Задание 2.

Исходные данные:

На входной двери подъезда установлен кодовый замок, содержащий десять кнопок с цифрами от 0 до 9. Код содержит три цифры, которые нужно нажать одновременно. Какова вероятность того, что человек, не знающий код, откроет дверь с первой попытки?

Решение:

```

Python 3.8.5 (default, May 27 2021, 13:30:53)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> from math import factorial
>>> a = 3
>>> b = 10
>>> c = a / b
>>> def result(c):
...     try:

```

```

...     a = int(input("Введите число: "))
...     b = int(input("Введите число: "))
... except zerodivisionerror:
...     return
... d = int(factorial(a))
... f = int(factorial(b))
... s = d / f
... return s
...
>>> print(result(c))
Введите число: 3
Введите число: 10
1.6534391534391535e-06
>>>

```

```

Python 3.8.5 (default, May 27 2021, 13:30:53)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> from math import factorial
>>> m = 10
>>> k = 3
>>> def combinations(m, k):
...     try:
...         m = int(input("Введите число: "))
...         k = int(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     s = int(factorial(m) / (factorial(k) * factorial(m - k)))
...     return s
...
>>> print(combinations(m,k))
Введите число: 10
Введите число: 3
120
>>>

```

Задание 3.

Исходные данные:

В ящике имеется 15 деталей, из которых 9 окрашены. Рабочий случайным образом извлекает 3 детали. Какова вероятность того, что все извлеченные детали окрашены?

Решение:

```

Python 3.8.5 (default, May 27 2021, 13:30:53)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> from math import factorial
>>> m = 15
>>> k = 3
>>> def combinations(m, k):
...     try:
...         m = int(input("Введите число: "))

```

```

...     k = int(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     s = int(factorial(m) / (factorial(k) * factorial(m - k)))
...     return s
...
>>> print(combinations(m, k))
Введите число: 15
Введите число: 3
455
>>>

```

Задание 4.

Исходные данные:

4. В лотерее 100 билетов. Из них 2 выигрышных. Какова вероятность того, что 2 приобретенных билета окажутся выигрышными?

Решение:

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> import math
>>> from math import factorial
>>> m = 100
>>> k = 2
>>> def combinations(m, k):
...     try:
...         m = int(input("Введите число: "))
...         k = int(input("Введите число: "))
...         except zerodivisionerror:
...             return
...         s = int(factorial(m) / (factorial(k) * factorial(m - k)))
...         return s
...
>>> print(combinations(m, k))
Введите число: 100
Введите число: 2
4950
>>>

```

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> import math
>>> from math import factorial
>>> m = 100
>>> k = 2
>>> def arrange(m, k):
...     try:
...         m = int(input("Введите число: "))
...         k = int(input("Введите число: "))
...         except zerodivisionerror:
...             return
...

```

```

...     s = int(factorial(m) / factorial(m - k))
...     return s
...
>>> print(arrange(m, k))
Введите число: 100
Введите число: 2
9900
>>>

```

Задание 5.

Исходные данные:

5. На соревновании по биатлону один из трех спортсменов стреляет и попадает в мишень. Вероятность попадания для первого спортсмена равна 0.9, для второго — 0.8, для третьего — 0.6. Найти вероятность того, что выстрел произведен: а). первым спортсменом б). вторым спортсменом в). третьим спортсменом.

Решение:

а). первым спортсменом.

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> a = 0.9
>>> b = 0.8
>>> c = 0.6
>>> def biatlon1(a):
...     try:
...         a = float(input("Введите число: "))
...         b = float(input("Введите число: "))
...         c = float(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     d = 1/3
...     f = (a * d) + (b * d) + (c * d)
...     s = a * d / f
...     return s
...

```

```

>>> print(biatlon1(a))

```

Введите число: 0.9

Введите число: 0.8

Введите число: 0.6

0.391304347826087

```

>>>

```

б). вторым спортсменом

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> a = 0.9
>>> b = 0.8
>>> c = 0.6
>>> def biatlon2(b):
...     try:
...         a = float(input("Введите число: "))

```

```

...     b = float(input("Введите число: "))
...     c = float(input("Введите число: "))
... except zerodivisionerror:
...     return
...     d = 1/3
...     f = (a * d) + (b * d) + (c * d)
...     s = b * d / f
...     return s
...
>>> print(biatlon2(b))
Введите число: 0.9
Введите число: 0.8
Введите число: 0.6
0.3478260869565218
>>>

```

в). третьим спортсменом

Python 3.8.5 (default, May 27 2021, 13:30:53)
 [GCC 9.3.0] on linux
 Type "help", "copyright", "credits" or "license" for more information.

```

>>> a = 0.9
>>> b = 0.8
>>> c = 0.6
>>> def biatlon3(c):
...     try:
...         a = float(input("Введите число: "))
...         b = float(input("Введите число: "))
...         c = float(input("Введите число: "))
...     except zerodivisionerror:
...         return
...     d = 1/3
...     f = (a * d) + (b * d) + (c * d)
...     s = c * d / f
...     return s
...
>>> print(biatlon3(c))
Введите число: 0.9
Введите число: 0.8
Введите число: 0.6
0.2608695652173913
>>>

```