

Задание 1.

Исходные данные:

Смоделируем \$60\$-кратное подбрасывание игральной кости с помощью функции `randint` из модуля `numpy.random`.

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 numpy as np
```

```
>>> n = 60
```

```
>>> a = np.random.randint(1, 7, size=n)
```

```
>>> print(a)
```

```
[1 4 2 6 2 4 6 1 5 5 2 2 5 2 3 2 3 5 4 1 3 2 2 5 1 3 6 2 3 3 4 2 3 1 6 1 6
```

```
1 2 2 4 3 4 6 1 4 1 4 5 6 4 1 6 4 1 6 3 5 6 5]
```

```
>>>
```

Событием B будем считать выпадение числа 3 . Сперва найдём его относительную частоту. Вычислим мощность подмножества, где наблюдалось событие B , т.е. выпадало число 3 :

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 = 60
```

```
>>> a = np.random.randint(1, 7, size=n)
```

```
>>> print(a)
```

```
[2 1 6 5 6 2 6 6 5 6 6 2 2 3 1 6 5 1 4 3 5 2 5 2 2 3 5 5 4 4 3 5 6 2 4 2 1
```

```
5 4 6 3 3 6 5 4 4 4 2 6 2 4 5 2 6 4 3 5 5 2 5]
```

```
>>>
```

```
>>> a == 3
```

```
array([False, False, False, False, False, False, False, False, False,
```

```
       False, False, False, False, True, False, False, False, False,
```

```
False, True, False, False, False, False, False, True, False,  
False, False, False, True, False, False, False, False, False,  
False, False, False, False, True, True, False, False, False,  
False, False, False, False, False, False, False, False, False,  
False, True, False, False, False, False])
```

```
>>>
```

```
>>> m = (a == 3).sum()
```

```
>>> print(m)
```

```
7
```

```
>>>
```

```
>>> w = m / n
```

```
>>> print(w)
```

```
0.11666666666666667
```

```
>>>
```

Задание 2.

Исходные данные:

Разберём более сложный пример. Смоделируем ситуацию, когда бросают две игральные кости одновременно.

При этом будем находить частоту случайного события B , при котором на первой кости выпало 1 , а на второй — 2 .

Проведём для этого 360 испытаний.

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 numpy as np
```

```
>>> n = 360
```

```
>>> c = np.random.randint(1, 7, size=n)
```

```
>>> d = np.random.randint(1, 7, size=n)
```

```
>>> print(c)
```

```
[5 2 1 4 5 2 5 3 5 3 1 6 3 4 5 1 1 3 1 3 2 4 6 4 1 3 3 1 4 1 3 6 1 4 5 4 4  
5 6 5 3 6 1 3 6 2 3 1 1 5 1 5 6 3 4 6 4 6 6 1 4 4 1 3 4 2 5 1 1 6 1 2 4 1
```

```
6455151262235161643251215256416514453
3632434346415156331454356643533614652
3643546154313531253423526465326445125
4424313535611323535245224544165544133
2414231164333314422221411261635644633
4363531443666556354412366514421331236
1563444236631326634351154652561351262
123664222363623523541344342]
```

```
>>>
```

```
>>> print(d)
```

```
[5216632654131426312256233216454543162
1153523156163642314456113563312125532
5561431546561212535213121516351311553
1364626422363415452352144441216633116
1361351654131136651543264464264414346
1114142362244434334466214262641554326
1514626536253454124125355661116642133
1113113362143312533122214623466244246
4541465636124444456352232461145544443
561645253344163363326112651]
```

```
>>> c[0]
```

```
5
```

```
>>> d[0]
```

```
5
```

```
>>>
```

```
>>> i = 0
```

```
>>> print(c[i], d[i])
```

```
5 5
```

```
>>> m = ((c == 1) & (d == 2)).sum()
```

```
>>> print(m)
```

```
12
```

```
>>> w = m / n
```

```
>>> print(w)
0.03333333333333333
>>>
```

Задание 3.

Исходные данные:

Сколькими способами можно выбрать из колоды, состоящей из 36 карт, 4 карты?

Поскольку здесь важен не порядок, а лишь содержание, воспользуемся формулой сочетаний:

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 = 36
>>> 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: 36
Введите число: 4
58905
>>>
```

Задание 4.

Исходные данные:

В магазине \$20\$ покупателей. Сколькими способами они могут образовать очередь из \$5\$ человек?

В этом примере важен порядок, в котором покупатели будут стоять в очереди, поэтому применим формулу размещений.

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 = 20
```

```
>>> k = 5
```

```
>>> 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: 20

Введите число: 5

1860480

```
>>>
```

Задание 5.

Исходные данные:

способами \$5\$ покупателей могут образовать очередь?

Этот пример похож на предыдущий, но есть важное отличие: не нужно выбирать \$5\$ покупателей из \$20\$\$. Здесь их всего \$5\$\$, и все должны присутствовать в очереди, поэтому нам нужно число перестановок.

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 = 5
```

```
>>> def permutations(m):
```

```
...     try:
```

```
...         m = int(input("Введите число: "))
```

```
...     except zerodivisionerror:
```

```
...         return
```

```
...     s = int(factorial(m))
```

```
...     return s
```

```
...
```

```
>>> print(permutations(m))
```

Введите число: 5

120

```
>>>
```

Задание 6.

Исходные данные:

Сколькими способами можно из колоды, состоящей из 36 карт, выбрать 5 так, чтобы среди них оказалось 2 туза?

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 = 4
```

```
>>> b = 2
```

```
>>> c = 32
```

```
>>> d = 3
```

```
>>> def combinations(a, b, c, d):
```

```
...     try:
```

```
...         a = int(input("Введите число: "))
```

```
...         b = int(input("Введите число: "))
```

```
...         c = int(input("Введите число: "))
```

```
...         d = int(input("Введите число: "))
```

```
...     except zerodivisionerror:
```

```
...         return
```

```
...         f = int(factorial(a) / (factorial(b) * factorial(a - b)))
```

```
...         g = int(factorial(c) / (factorial(d) * factorial(c - d)))
```

```
... s = f * g
```

```
... return s
```

```
...
```

```
>>> print(combinations(a, b, c, d))
```

Введите число: 4

Введите число: 2

Введите число: 32

Введите число: 3

29760

```
>>>
```

Задание 7.

Исходные данные:

Сколькими способами можно из колоды, состоящей из 36 карт, выбрать 5 так, чтобы среди них оказалось от 2 до 3 тузов?

В чём отличие этой задачи от предыдущей? В предыдущей считались только наборы из 5 карт с двумя тузами. Здесь же тузов может быть как 2, так и 3.

Удобнее всего эту задачу решать по частям. Мы уже нашли число a способов достать 5 карт, среди которых ровно 2 туза. Найдём аналогичным образом число b способов достать 5 карт, среди которых ровно 3 туза.

Решение:

Первое значение мы берём из задания 6.

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 = 4

>>> b = 3

>>> c = 32

>>> d = 2

>>> def combinations(a, b, c, d):

...     try:

...         a = int(input("Введите число: "))

...         b = int(input("Введите число: "))

...         c = int(input("Введите число: "))

...         d = int(input("Введите число: "))

...     except zerodivisionerror:

...         return

...     f = int(factorial(a) / (factorial(b) * factorial(a - b)))

...     g = int(factorial(c) / (factorial(d) * factorial(c - d)))

...     s = f * g

...     return s

...

>>> print(combinations(a, b, c, d))

Введите число: 4

Введите число: 3

Введите число: 32

Введите число: 2

1984

>>>
```

$$k = 29760 + 1984$$

$$k = 31744$$

Задание 8.

Исходные данные:

Вычислим вероятность того, что на игральной кости выпадет число 3\$. Известно, что у кости шесть граней с числами от 1\$ до 6\$ и каждая может выпасть с равной вероятностью. По классической формуле вероятности получаем.

Решение:

Так как у нас тройка есть только одно число из шести.

Следовательно делаем по формуле.

$$A = 1$$

$$B = 6$$

$$P = A / B$$

$$P = 1 / 6$$

$$P = 1/6=0.16666666666666666$$

Задание 9.

Исходные данные:

Бросаются одновременно три игральных кубика. Какова вероятность того, что все три значения будут одинаковыми?

Всего есть $6 \cdot 6 \cdot 6$ исходов, поскольку каждый из кубиков может выпасть любой из 6\$ сторон. Нас интересуют исходы, при которых все три значения совпали. Таких событий 6\$.

Решение:

$$A = 6$$

$$B = 6 * 6 * 6$$

$$P = A / B$$

$$P = 6 / (6 * 6 * 6)$$

$$P = 1 / 36$$

Задание 10.

Исходные данные:

Наугад выбирается точка с круглой мишени. Пусть событие A заключается в выборе точки из правой половины, а событие B — выбор точки из нижней половины. Являются ли эти события зависимыми?

Решение:

Так как у нас точки попадания имеют вертикальное расположение, точнее расположены в разных четвертях, если смотреть по графику, то получаем следующий результат.

$$A = \frac{1}{4} / \frac{1}{2} = \frac{1}{2}$$

и вторая точка получается тоже самое.

$$B = \frac{1}{4} / \frac{1}{2} = \frac{1}{2}$$

Следовательно, данные точки независимы и соответственно они равны.

$$A = B$$

Задание 11.

Исходные данные:

Есть три одинаковые корзины. В первой лежат три красных и пять зелёных мячей, во второй — только красные, а в третьей — только зелёные мячи. Случайно выбирается одна корзина, и из неё случайным образом извлекается мяч.

Вопрос: Какова вероятность того, что этот мяч — зелёный?

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
```

```
>>> a1 = 8
```

```

>>> a2 = 5
>>> a3 = 5
>>> b1 = 3
>>> b2 = 5
>>> b3 = 0
>>> c1 = 5
>>> c2 = 0
>>> c3 = 5
>>> def func(a1, b1, c1):
...     try:
...         a1 = int(input("Введите число: "))
...         b1 = int(input("Введите количество красных мячей: "))
...         c1 = int(input("Введите количество зелёных мячей: "))
...         a2 = int(input("Введите число: "))
...         b2 = int(input("Введите количество красных мячей: "))
...         c2 = int(input("Введите количество зелёных мячей: "))
...         a3 = int(input("Введите число: "))
...         b3 = int(input("Введите количество красных мячей: "))
...         c3 = int(input("Введите количество зелёных мячей: "))
...     except zerodivisionerror:
...         return
...     d = c1 / a1
...     f = 1 / 3
...     s = (d * f) + (c2 * f) + (c3 * f)
...     return s
...
>>> print(func(a1, b1, c1))
Введите число: 8
Введите количество красных мячей: 3
Введите количество зелёных мячей: 5
Введите число: 5
Введите количество красных мячей: 5
Введите количество зелёных мячей: 0
Введите число: 5
Введите количество красных мячей: 0
Введите количество зелёных мячей: 5
1.8749999999999998
>>>

```

Задание 12

Исходные данные:

Продолжим задание 11: таким же образом выбирается наугад корзина, и из неё случайным образом извлекается мяч. Допустим, мяч оказался зелёным. Какова вероятность, что он извлекался: а) из первой корзины, б) из второй корзины? в) из третьей корзины?

Решение:

а) из первой корзины

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 = 0.0625
```

```
>>> b = 0
```

```
>>> c = 5
```

```
>>> def bask1(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(bask1(a))
```

Введите число: 0.0625

Введите число: 0

Введите число: 5

0.01234567901234568

```
>>>
```

б) из второй корзины

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 = 0.0625
```

```
>>> b = 0
```

```
>>> c = 5
```

```
>>> def bask2(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(bask2(b))
```

Введите число: 0.0625

Введите число: 0

Введите число: 5

0.0

```
>>>
```

в) из третьей корзины

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 = 0.0625
```

```
>>> b = 0
```

```
>>> c = 5
```

```
>>> def bask3(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(bask3(c))
```

Введите число: 0.0625

Введите число: 0

Введите число: 5

0.9876543209876544

```
>>>
```

Задание 13.

Исходные данные:

Соревнования по биатлону. Один из трёх спортсменов стреляет и попадает в мишень. Вероятность такого события для первого спортсмена равна \$0.2\$, для второго — \$0.4\$, для третьего — \$0.7\$.

Задача: найти вероятность того, что стрелял третий спортсмен.

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.

```
>>> a = 0.2
```

```
>>> b = 0.4
```

```
>>> c = 0.7
```

```
>>> def biatlon(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(biatlon(c))
```

Введите число: 0.2

Введите число: 0.4

Введите число: 0.7

0.5384615384615384

```
>>>
```