# ЛАБОРАТОРНАЯ РАБОТА 5.
# СПИСОК (МАССИВ). ОПЕРАЦИИ СО СПИСКАМИ В PYTHON

## 5.1. СПРАВОЧНЫЙ МАТЕРИАЛ

**Массив** – это структура данных, доступ к элементам которой осуществляется по номеру (индексу) или по нескольким индексам (для многомерных массивов). Все элементы ***в массиве имеют одинаковый тип***. Наименьший индекс массива равен 0. Однако, если подходить формально, в языке Python такая структура как массив отсутствует. Вместо нее используются списки. **Списки в Python** — это определенное количество элементов, которые имеют общее имя, и каждый элемент имеет свой индекс — порядковый номер (или несколько индексов, если у списка несколько измерений). В отличие от строгих массивов, у списков ***нет никаких ограничений на тип переменных***, поэтому ***в списках могут храниться объекты разного типа***. Частным случаем этого, является как раз список, где все элементы одного типа, как в обычных массивах. Например, можно задать список L – целых чисел так (или назвать это массив целых чисел):

```
L = [24, 765, -48, 59, -42]
```

Списки являются упорядоченными последовательностями, которые состоят из различных объектов (значений, данных), заключающихся в квадратные скобки [ ] и отделяющиеся друг от друга с помощью запятой. Элементы списка могут изменяться в процессе работы программы.

Наиболее полно работа с формальными массивами для языка Python реализована в библиотеке **Numpy**, которая содержит различные числовые типы данных, многомерные массивы, матричные структуры данных и широко используется для выполнения множества математических и статистических операций с массивами.

### Создание списков на Python

Создать список можно несколькими способами.

1. Получение списка через присваивание конкретных значений. Пример в коде Python пустой список:

```
s = []  # Пустой список
```

Примеры создания списков со значениями: Примеры списков:

```
l = [25, 755, -40, 57, -41]   # список целых чисел
l = [1.13, 5.34, 12.63, 4.6, 34.0, 12.8]   # список из дробных чисел
l = ["Sveta", "Sergei", "Ivan", "Dasha"]   # список из строк
l = ["Москва", "Иванов", 12, 124]   # смешанный список
l = [[0, 0, 0], [1, 0, 1], [1, 1, 0]]   # список, состоящий из списков
l = ['s', 'p', ['isok'], 2] # список из значений и списка
```

2. Списки при помощи функции List()

```
empty_list = list() # пустой список
l = list ('spisok')  # 'spisok' - строка
print(l) #['s', 'p', 'i', 's', 'o', 'k'] - результат - список
```

3. Создание списка при помощи функции Split()

Используя функцию **split** в Python можно получить из строки список. Рассмотрим пример:

```
stroka ="Hello, world"  # stroka - строка
lst=stroka.split(",") # lst - список
lst # ['Hello', ' world']
```

**Таблица. Основные методы списков (a = [] – список с элементами)**

| Метод | Описание (что делает) | Пример |
|---|---|---|
| **x in a** | Проверка, что x содержится в a | 5 in [2, 3, 5] |
| **x notin a** | Проверка, что x не содержится в a То же, что и not (x in a) | 5 notin [2, 3, 6] |
| **a + a2** | Конкатенация списков, то есть новый список, в котором сначала идут все элементы a, а затем все элементы a2 | [2, 4] + [5, 3] == [2, 4, 5, 3] |
| **a * k** | Список a, повторенный k раз | [2, 3] * 3 == [2, 3, 2, 3, 2, 3] |
| **a[n]** | n-й элемент списка, отрицательные n — для отсчёта с конца | [2, 3, 7][0] == 2<br>[2, 3, 7][-1] == 7 |
| **a[start:stop:step]** | Срез списка | [2, 3, 7][:2] == [2, 3] |
| **len(a)** | Длина списка | len([2, 3, 7]) == 3 |
| **max(a)** | Максимальный элемент списка | max([2, 3, 7]) == 7 |
| **min(a)** | Минимальный элемент списка | min([2, 3, 7]) == 2 |
| **sum(a)** | Сумма элементов списка | sum([2, 3, 7]) == 12 |
| **a.append**(x) | Добавляет элемент в конец списка | |
| **a.extend**(L) | Расширяет список list, добавляя в конец все элементы списка L | |
| **a.insert**(i, x) | Вставляет перед i-ым элементом значение x | |
| **a.remove**(x) | Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует | |
| **a.pop**([i]) | Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент | |

| | | |
|---|---|---|
| **a.index**(x, [start [, end]]) | Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end) | |
| **a.count**(x) | Возвращает количество элементов со значением x | |
| **a.reverse**() | Разворачивает список | |
| **a.copy**() | Поверхностная копия списка | |
| **a.clear**() | Очищает список | |
| **del a[n]** | Удалить n-й элемент списка | a = [2, 3, 7] del a[1] a == [2, 7] |
| **Del a[start:stop:step]** | Удалить из a все элементы, попавшие в срез | a = [2, 3, 7] del a[:2] a == [7] |
| **a.clear()** | Удалить из a все элементы (то же, что del a[:]) | a.clear() |
| **a.copy()** | Копия a (то же, что и полный срез a[:]) | b = a.copy() |
| **a += a2** **a *= k** | Заменить содержимое списка на a+$a_2$ и a*k соответственно | |
| **a.reverse()** | Изменить порядок элементов в a на обратный (перевернуть список) | a = [2, 3, 7] a.reverse() a == [7, 3, 2] |
| **a.sort()** | Отсортировать список по возрастанию | a = [3, 2, 7] a.sort() a == [2, 3, 7] |
| **a.sort(reverse=True)** | Отсортировать список по убыванию | a = [3, 2, 7] a.sort(reverse = True) a == [7, 3, 2] |
| **bool(a)** | Один из способов проверить список на пустоту (возвращает True, если список непустой, и False в противном случае) | |

**Двумерные и многомерные списки (массивы)**

Двумерный список(массив) можно представить, как список, элементами которого являются другие списки, т.е. это своего рода «список списков». Такие виды списков (массивов) называют двумерными списками (массивами), матрицами или таблицами. Доступ к элементам матрицы осуществляется с помощью указания двух индексов: индекс строки матрицы и индекс столбца матрицы. Можно рассматривать их также как индекс подсписка внутри списка и индекс элемента внутри этого подсписка.

Для примера создадим таблицу с тремя столбцами и тремя строками, заполненными произвольными буквами:

```
mas = [['й', 'ц', 'у'], ['к','е','н'], ['г', 'ш', 'щ']]
#Вывод всего двумерного массива
print(mas)
```

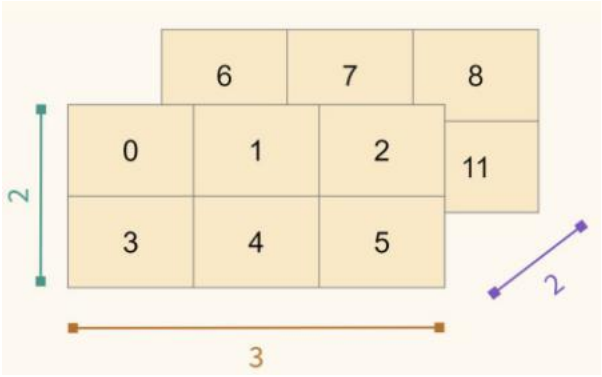
Результат:

```
[['й', 'ц', 'у'], ['к', 'е', 'н'], ['г', 'ш', 'щ']]
```

```
#Вывод первого элемента в первой строке
print(mas[0][0]) # Выведет й
#Вывод третьего элемента в третьей строке
print(mas[2][2]) # Выведет щ
```

Трехмерный список можно рассматривать как список двумерных списков.
Четырехмерный список – как список трехмерных списков и так далее.
N-мерный список – это список (N–1)-мерных списков. Соответственно количество индексов у N-мерного списка будет ровно N.

Например, создадим трехмерный массив с размерностями 2x2x3:




```
mas = ([[[ 0,  1,  2],
         [ 3,  4,  5]],
        [[ 6,  7,  8],
         [ 9, 10, 11]]])
```

Выведем некоторые элементы этого массива:

```
>>> print (mas[1][1][1])
10
>>> print (mas[0][0][0])
0
>>> print (mas[0][1][2])
5
```

**Создание двумерных массивов**

Создать такой массив в Python можно разными способами.

**Первый способ.**

```
# Создание таблицы с размером a x b, заполненной нулями
a = 3
b = 4
mas = [0] * a
for i in range(a):
```

```
    mas[i] = [0] * b
print(mas)
# Выведет [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
```

**Второй способ** предполагает создание пустого списка с добавлением в него новых списков. Рассмотрим на примере:

```
# Создание таблицы с размером a x b, заполненной единицами
a = 3
b = 3
mas = []
for i in range(a):
    mas.append([1] * b)
print(mas) # Выведет [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
```

**Третьим и самым простым способом** является генератор списков с **X** строками, которые будут состоять из **Y** элементов (для матрицы X на Y). Пример:

```
# Создание таблицы с размером X x Y, заполненной двойками
X = 4
Y = 3
mas = [[2] * Y for i in range(X)]
print(mas)
# Выведет [[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
```

**Четвертый способ. Заполнение массива с помощью генератора случайных чисел.**

Будем использовать библиотеку **random** и функцию из нее **randint** (a,b), которая генерирует случайное целое число в диапазоне [a,b]. Сгенерируем массив целых чисел в диапазоне от 0 до 100 размерностью 4 строки на 5 столбцов (4x5):

```
import random
x=4
y=5
array = [[random.randint(0, 100) for i in range(y)] for j in range(x)]
```

Результат:

```
>>> array
[[39, 96, 77, 59, 6], [75, 96, 98, 92, 87], [31, 31, 30, 41, 18], [41, 86, 83, 23, 27]]
```

**Способы ввода двумерных массивов**

Допустим, нам нужно ввести двумерный массив после запуска нашей программы. Для этого мы можем создать программу, которая будет построчно

считывать значения нашего массива, а также количество строк в нем.

Таким образом, количество строк в массиве считывается в начале, а потом в каждой строке размещается нужное количество элементов через пробел.

Однако следует учитывать, что ввод данных осуществляется в виде строки, ее нужно разобрать и потом введенные целые числа преобразовать в тип int.

Рассмотрим на примере:

```
a=int(input("Введите количество срок в массиве:"))
mas = []
for i in range(a):
    mas.append(list(map(int, input().split())))
print(mas)
```

Запускаем программу и сначала вводим количество строк в массиве (допустим, 3). Далее вводим строки в порядке их очереди. Например:

```
1 1 1
1 1 1
1 1 1
```

После этого данная программа выведет наш двумерный массив:

```
[[1, 1, 1], [1, 1, 1], [1, 1, 1]].
```

То же самое можно сделать с помощью генератора двумерных массивов:

```
mas = [list(map(int, input().split())) for i in range(int(input()))]
# Вводим
3
1 1 1
1 1 1
1 1 1
print(mas) # Выведет [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
```

**Функция map для массивов**

Здесь использовалась функция **map** для массива, которая в общем случае выглядит так:

```
map(function, iterable, ...)
```

Параметры:

- function – пользовательская функция, вызывается для каждого элемента,
- iterable – последовательность или объект, поддерживающий итерирование.

Возвращаемое значение: **map object** – объект итератора.

Функция map() выполняет пользовательскую функцию function для каждого элемента последовательности, коллекции или итератора iterable. Каждый элемент iterable отправляется в функцию function в качестве

аргумента. Если в map() передаётся несколько iterable, то пользовательская функция function должна принимать количество аргументов, соответствующее количеству переданных последовательностей, при этом function будет применяться к элементам из всех итераций параллельно.

В нашем случае в функцию map передается параметр int – это функция преобразования строки в целое число. При этом input().split() выделяет из входной строки следующее число, разделенное пробелами.

**Вывод двумерных массивов**

Для обработки и вывода списков используются два вложенных цикла. Первый цикл – по порядковому номеру строки, второй – по ее элементам. Например, вывести массив можно так:

```
mas = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
for i in range(0, len(mas)):
    for i2 in range(0, len(mas[i])):
        print(mas[i][i2], end=' ')
    print()
# Выведет
1 1 1
1 1 1
1 1 1
```

То же самое можно сделать не по индексам, а по значениям массива:

```
mas = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
for i in mas:
    for i2 in i:
        print(i2, end=' ')
    print()
# Выведет
1 1 1
1 1 1
1 1 1
```

Способ с использованием метода join():

```
mas = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
for i in mas:
    print(' '.join(list(map(str, i))))
# Выведет
1 1 1
1 1 1
1 1 1
```

Вывод одной из строк двумерного массива можно осуществить с помощью цикла и того же метода join(). Для примера выведем вторую строку в произвольном двумерном массиве:

```
mas = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
string = 2
for i in mas[string-1]:
    print(i, end=' ')
# Выведет 1 1 1
```

Для вывода определенного столбца в двумерном массиве можно использовать такую программу:

```
mas = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
column = 2
for i in mas:
    print(i[column-1], end=' ')
# Выведет 2 5 8
```

**Обработка двумерных массивов**

Составим произвольный двумерный массив с числами и размерностью

```
2 4 7 3
4 5 6 9
1 0 4 2
7 8 4 7
```

Теперь поставим числа в каждой строке по порядку:

```
mas = [[2, 4, 7, 3], [4, 5, 6, 9], [1, 0, 4, 2], [7, 8, 4, 7]]
mas2 = []
for i in mas:
    mas2.append(sorted(i))
print(mas2)
# Выведет [[2, 3, 4, 7], [4, 5, 6, 9], [0, 1, 2, 4], [4, 7, 7, 8]]
```

А теперь расставим все числа по порядку, вне зависимости от их нахождения в определенной строке:

```
mas = [[2, 4, 7, 3], [4, 5, 6, 9], [1, 0, 4, 2], [7, 8, 4, 7]]
mas2 = []
for i in mas:
    for i2 in i:
        mas2.append(i2)
mas=sorted(mas2)
for x in range(0, len(mas), 4):
```

```
        e_c = mas[x : 4 + x]
        if len(e_c) < 4:
            e_c = e_c + [None for y in range(n - len(e_c))]
        print(list(e_c))
    # Выведет
    [0, 1, 2, 2]
    [3, 4, 4, 4]
    [4, 5, 6, 7]
    [7, 7, 8, 9]
```

## 5.2. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Генераторы списков:

**Пример 1:**

# список из 10 элементов, заполненный единицами

L

# список L = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

**Пример 2:**

L = [i for i in range(10)]

# список L = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

**Пример 3:**

L = [i*2 for i in range(10)]

# список L = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

**Пример 4:** добавим условие для выбора только четных чисел из примера

L = [i*2 for i in range(10) if i % 2 == 0]

# список L = [0, 4, 8, 12, 16]

**Пример 5. Пример работы со строкой:**

c = [c * 3 for c in 'list']

print (c) # ['lll', 'iii', 'sss', 'ttt']

**Пример 6**. Ввод элементов списка, задаваемого пользователем с помощью цикла.

```
N = int(input("Введите количество элементов списка! "))
L = [int(input("Введите элемент")) for i in range(N)]
print(L)
```

## 5. 3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### Задачи для различных вариантов (список вариантов см. в конце руководства).

1. Массив **A(20)** задается с помощью датчика случайных чисел (от 1 до 100). Вывести массив и найти максимальный элемент массива.
2. Массив **N(10**) задается с помощью датчика случайных чисел. Вывести массив на экран. Распечатать первый четный элемент массива.
3. Массив **A(20**) задается с помощью датчика случайных чисел. Вывести массив на экран. Посчитать сколько четных и сколько нечетных элементов в массиве.
4. Массив **M(15)** задается с помощью датчика случайных чисел. Вывести массив на экран. Определить сумму его элементов. Напечатать также минимальный элемент массива.
5. Массив **B(20)** задается с помощью датчика случайных чисел (от -50 до 50). Вывести массив. Подсчитать количество отрицательных, положительных чисел и нулей.
6. Массив **A(15)** задается с помощью датчика случайных чисел (от 1 до 100). Все четные элементы поместить в массив L, а нечетные — в массив F. Все массивы вывести на экран.
7. Массив **A(20)** задается с помощью датчика случайных чисел (от 1 до 100). Подсчитать: сколько чисел больше 50 и сколько чисел меньше 50. А также подсчитать: сколько четных чисел и сколько – нечетных.
8. Массив **M(15)** задается с помощью датчика случайных чисел. Вывести массив на экран. Найти минимальный и максимальный элементы массива и поменять их местами. Вывести массив после преобразования.
9. Дан одномерный массив целых чисел **A(20)** (массив задается с помощью датчика случайных чисел)**.** Среди них есть четные числа. Вывести на экран массив с номерами четных элементов.
10. Массив **M(15)** задается с помощью датчика случайных чисел. Найти сумму трех минимальных элементов.
11. Двумерный массив целых чисел **A(M, N)** задается с помощью датчика случайных чисел. **M** и **N** – вводятся с экрана. Ко всем четным элементам массива прибавить цифру 5. Все нечетные элементы заменить нулями. Вывести массивы до и после преобразований.
12. Двумерный массив целых чисел **B(M, N)** задается с помощью датчика случайных чисел. Вывести массив на экран. Найти наибольший элемент массива. Напечатать также его индексы.
13. Дан двумерный массив целых чисел **A(M, N)** (массив задается с помощью датчика случайных чисел). **M** и **N** – вводятся с экрана. Ко всем четным элементам массива прибавить первый элемент соответствующей строки. Все элементы массива, оканчивающиеся цифрой 2, умножить на последний элемент соответствующего столбца.
14. Дан двумерный массив целых чисел **A(10,10)** (массив задается с помощью датчика случайных чисел). Просуммировать элементы массива по каждой из линий, параллельных главной диагонали. Напечатать полученные суммы.
15. Двумерный массив целых чисел **A(5,5)** задается с помощью датчика случайных чисел. Минимальный элемент пятой строки прибавить к каждому элементу первого столбца. Вывести массив до и после преобразований.

16. Квадратная матрица порядка **N** задается с помощью датчика случайных чисел. Вычислить среднее арифметическое элементов, стоящих на главной диагонали, предварительно отсортировав их.

17. Двумерный массив целых чисел **A(M, N)** задается с помощью датчика случайных чисел от -50 до +50. **M** и **N** – вводятся с экрана. Вывести количество строк матрицы, в которых число положительных элементов больше числа отрицательных элементов.

18. Двумерный массив целых чисел **B(M, N)** задается с помощью датчика случайных чисел. **M** и **N** – вводятся с экрана. Вывести массив на экран. Отсортировать элементы предпоследнего столбца по возрастанию, а затем элементы второй строки по убыванию.

19. Двумерный массив целых чисел **B(M, N)** задается с помощью датчика случайных чисел. **M** и **N** – вводятся с экрана. Вывести массив на экран. Найти произведение минимального элемента первого столбца и максимального элемента последней строки.

20. Дан двумерный массив целых чисел **A(M, N)** (массив задается с помощью датчика случайных чисел). **M** и **N** – вводятся с экрана. Вычислить количество столбцов матрицы, в которых нет ни одного четного элемента. Распечатать также их номера.

21. Создайте список случайных целых чисел от -20 до 30.

22. Создайте список целых чисел от -10 до 10 с шагом 2 (генерация list).

23. Создайте список из сумм троек чисел от 0 до 10, используя генератор списка (0 + 1 + 2, 1 + 2 + 3, …).

24. Заполните массив элементами арифметической прогрессии. Её первый элемент, разность и количество элементов нужно ввести с клавиатуры.* Формула для получения n-го члена прогрессии: **a**n **= a**1 **+ (n-1) * d**

25. В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.

26. Для списка A, состоящего из 20 элементов, вычислить количество отрицательных элементов списка.

27. Для списка B, состоящего из 20 элементов, вычислить сумму положительных элементов списка.

28. Для последовательности, состоящей из 16 элементов, создать список из значений под четными номерами этой последовательности.

29. Для списка X, состоящего из 15 элементов, найти наибольший элемент списка и его порядковый номер.

30. Для списка Y, состоящего из 15 элементов, найти наименьший элемент списка и его порядковый номер.

31. Для списка Z, состоящего из 20 элементов, найти наименьший из положительных элементов.

32. Для списка X, состоящего из 14 элементов, вывести на печать номера элементов, удовлетворяющих условию $0 < X(i) < 1$.

33. Информация о температуре воздуха за март задана в виде списка. Определить сколько раз температура опускалась ниже 0° C.

34. Дана последовательность целых чисел a1, a2, …, a18. Создать список из четных чисел этой последовательности. Если таких чисел нет, то вывести сообщение об этом факте.

35. При поступлении в вуз абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. В списке A записаны оценки 25 экзаменующихся, полученные на первом экзамене. Подсчитать, сколько человек не допущено ко второму экзамену.

36. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.

37. Дан произвольный список. Представьте его в обратном порядке.

38. Николай задумался о поиске «бесполезного» числа на основании списка. Суть оного в следующем: он берет произвольный список чисел, находит самое большое из них, а затем делит его на длину списка. Студент пока не придумал, где может пригодиться подобное значение, но ищет у вас помощи в реализации такой функции useless(s).

39. Создайте список из имен-строк. Затем добавьте в этот список два новых имени и удалите последнее имя. Выведите финальный список на консоль.

40. Пусть дан список, который содержит три списка (представить самим). Напишите программу, которая удаляет дубликаты из списка.

41. Напишите программу, которая с помощью цикла создает список чисел от 1 до 10, а также списки их квадратов и кубов. В конце списки выводятся на консоль.

42. Напишите программу, которая удаляет из списка чисел четные числа.

43. Дан список некоторых целых чисел, найдите значение 10 в нем и, если оно присутствует, замените его на 100. Обновите список только при первом вхождении числа 10.

44. Напишите программу, чтобы найти все значения в списке больше указанного числа.

45. Напишите программу, чтобы проверить, существует ли n-й элемент в данном списке.

46. Напишите программу, чтобы проверить, все ли элементы списка равны заданной строке.

47. Напишите программу, чтобы разбить список на разные переменные.

48. Напишите программу для разделения списка на основе первого символа слова.

49. Напишите программу на Python для преобразования списка из нескольких целых чисел в одно целое число.

50. Напишите программу для создания списка путем конкатенации заданного списка, диапазон которого варьируется от 1 до n. Пример списка: ['ш', 'в'] n = 5. Пример вывода: ['ш1', 'в1', 'ш2', 'в2', 'ш3', 'в3', 'ш4', 'в4', 'ш5', 'в5']

## 4.4. ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ ДОЛЖЕН ВКЛЮЧАТЬ

1. Титульный лист по форме с номером варианта.
2. Для каждой задачи по вариантам, всего 6 задач:
    - условие задачи;
    - блок-схема алгоритма;
    - программный код решения этой задачи (листинг);
    - скриншоты выполнения программы.

---

***Внимание!*** Отчет должен быть набран шрифтом **Times New Roman** и отформатирован: поля: левое – 3,5; правое – 1,5; нижнее и верхнее – 2 см; красная строка (отступ) - 1 см; межстрочный интервал – одинарный; правый край выровнен по ширине; рисунки сопровождены надписями.

## Распределение задач по вариантам

| № задачи | Вариант 1 | Вариант 2 | Вариант 3 | Вариант 4 | Вариант 5 | Вариант 6 | Вариант 7 | Вариант 8 | Вариант 9 | Вариант 10 | Вариант 11 | Вариант 12 | Вариант 13 | Вариант 14 | Вариант 15 | Вариант 16 | Вариант 17 | Вариант 18 | Вариант 19 | Вариант 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | * | | | | | | | * | | | | * | | | | | | | * | |
| 2. | | * | | | | | | | * | | | | * | | | | | * | | |
| 3. | | | * | | | | | * | | * | | | | * | | | | | | |
| 4. | | | | * | | | | | * | | * | | | | * | | | | | |
| 5. | * | | | | * | | | | | | | * | | | | * | | | | |
| 6. | | * | | | | * | | | | | | | * | | | | * | | | |
| 7. | | | * | | | | * | | | | | | | * | | | | * | | |
| 8. | | | | * | | | | * | | | | | | | * | | | | * | |
| 9. | | | | | * | | | | * | | | | | | | * | | | | * |
| 10. | * | | | | | * | | | | * | | | | | | | * | | | |
| 11. | | * | | | | | * | | | | * | | | | | | | * | | |
| 12. | | | * | | | | | | | | | * | | | | * | | | * | |
| 13. | | | | * | | | | | | | | | * | | | | * | | | * |
| 14. | | | | | * | | | | | * | | | | * | | | | * | | |
| 15. | | | | | | * | | | | | * | | | | * | | | | * | |
| 16. | | | | | * | | * | | | | | | | | | * | | | | * |
| 17. | * | | | | | | | * | | | | * | | | | | * | | | |
| 18. | | * | | | | * | | | * | | | | * | | | | | | | |
| 19. | | | * | | | | * | | | * | | | | * | | | | | | |
| 20. | | | | * | | | | | | | * | | | | * | | | | | * |
| 21. | * | | | | | | | | | | | | | | | | | | | |
| 22. | | * | | | | | | | | | | | | | | | | | | |
| 23. | | | * | | | | | | | | | | | | | | | | | |
| 24. | | | | * | | | | | | | | | | | | | | | | |
| 25. | | | | | * | | | | | | | | | | | | | | | |
| 26. | | | | | | * | | | | | | | | | | | | | | |
| 27. | | | | | | | * | | | | | | | | | | | | | |
| 28. | | | | | | | | * | | | | | | | | | | | | |
| 29. | | | | | | | | | * | | | | | | | | | | | |
| 30. | | | | | | | | | | * | | | | | | | | | | * |
| 31. | | | | | | | | | | | * | | | | | | | | * | |
| 32. | | | | | | | | | | | | * | | | | | | * | | |
| 33. | | | | | | | | | | | | | * | | | | * | | | |
| 34. | | | | | | | | | | | | | | * | | * | | | | |
| 35. | | | | | | | | | | | | | | | * | | | | | |
| 36. | * | | | | | | | | | | | | | | | | | | | |
| 37. | | * | | | | | | | | | | | | | | | | | | |
| 38. | | | * | | | | | | | | | | | | | | | | | |
| 39. | | | | * | | | | | | | | | | | | | | | | |
| 40. | | | | | * | | | | | | | | | | | | | | | |
| 41. | | | | | | * | | | | | | | | | | | | | | |
| 42. | | | | | | | * | | | | | | | | | | | | | |
| 43. | | | | | | | | * | | | | | | | | | | | | |
| 44. | | | | | | | | | * | | | | | | | | | | | |
| 45. | | | | | | | | | | * | | | | | | | | | | * |
| 46. | | | | | | | | | | | * | | | | | | | | * | |
| 47. | | | | | | | | | | | | * | | | | | | * | | |
| 48. | | | | | | | | | | | | | * | | | | * | | | |
| 49. | | | | | | | | | | | | | | * | | * | | | | |
| 50. | | | | | | | | | | | | | | | * | | | | | |