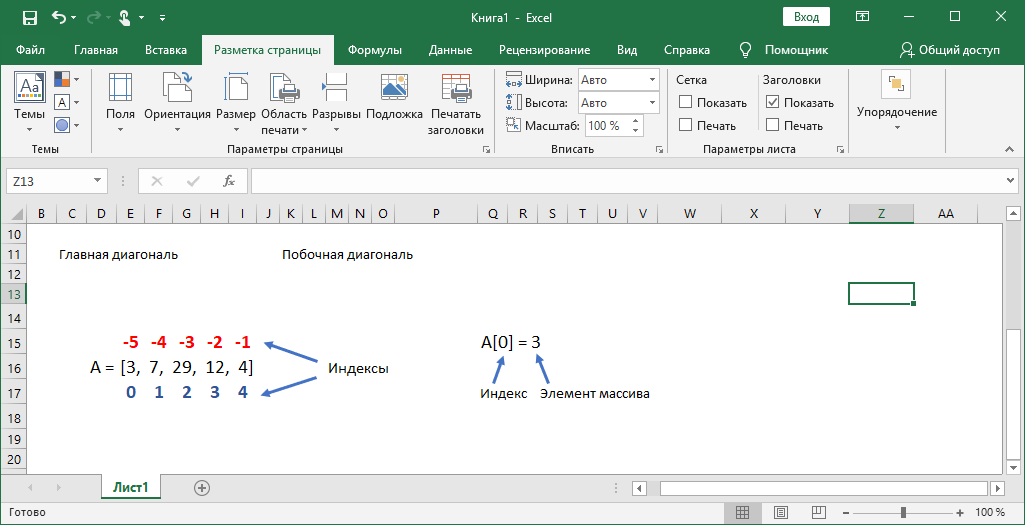
Массивы (списки)

Индексы и элементы массива (списка):



|  |  |
| --- | --- |
| **Формулировка задачи** | **Код** |
| **Ввод массива (списка)** | |
| На вход программы подается n - количество элементов в массиве, затем сами элементы в строке через пробел  5  3 4 5 6 3 | n = int(input()) # считываем количество элементов в массиве  A = list (map (int, input().split())) # считываем строку, разбиваем по пробелам,  # преобразуем в целые числа, затем получаем  # массив |
| На вход программы подается n - количество элементов в массиве, затем сами элементы по одному в строке  3  7  4  5 | A = [ ] # создаем пустой массив А  n = int(input()) # считываем количество элементов в массиве  for i in range (n):  A.append(int(input()) # n раз считываем очередной элемент и  # добавляем его в массив А |
| На вход программы подается n - количество элементов в массиве, затем сами элементы по одному в строке  3  7  4  5 | n = int(input()) # считываем количество элементов в массиве  A = [0] \* n # создаем массив А, состоящий из n нулей  for i in range (n):  A[i] = int(input() # n раз считываем очередной элемент и  # присваиваем его вместо A[i] |
| На вход программы подается n - количество элементов в массиве, затем сами элементы по одному в строке  3  7  4  5  Считать массив с помощью генератора | A = [ int(input()) for i in range (int(input()))]  Или n считать в отдельную переменную:  n = int(input())  A = [ int(input()) for i in range (n)] |
| **Создание массива (списка) с помощью генератора** | |
| Создать список, состоящий из n нулей при помощи генератора | A = [ 0 for i in range (n)] |
| Заполнить список квадратами чисел от 1 до n:  A = [1, 4, 9, 16, 25] | A = [ i \*\* 2 for i in range (1, n + 1)] |
| Получить список, заполненный случайными числами от 1 до 10 включительно (используя функцию randint из модуля random) | from random import \*  A = [ randint(1, 9) for i in range (n) ] |
| **Обработка массива (списка)** | |
| Вывести на экран все четные элементы массива А. | for i in range (len(A)): # перебираем индексы элементов  if A[i] % 2 == 0: # проверяем элемент массива на четность  print (A[i]) # если элемент четный, то печатаем его |
| Вывести на экран все четные элементы массива А, перебирая элементы, минуя индексы. | for elem in A: # перебираем элементы массива А  if elem % 2 == 0: # проверяем элемент массива на четность  print (elem) # если элемент четный, то печатаем его |
| **Вывод массива (списка) на экран** | |
| Вывести элементы массива А:  A = [2, 4, 6, 5] | print (A) # массив выведется со скобками и с запятыми: [2, 4, 6, 5] |
| Вывести элементы массива А:  A = [2, 4, 6, 5] | print (\*A) # массив выведется в виде: 2 4 6 5.  # Недостаток: медленная работа данной команды |
| Вывести элементы массива А:  A = [2, 4, 6, 5] | for i in range (len(A)): # перебираем индексы элементов  print (A[i], end = ‘ ‘) # выводим очередной элемент массива, затем пробел |
| Вывести элементы массива А:  A = [2, 4, 6, 5] | for elem in A: # перебираем элементы массива, минуя индексы  print (elem, end = ‘ ‘) # выводим очередной элемент массива, затем пробел |
| Вывести элементы строкового массива:  A = [‘2’, ‘4’, ‘6’, ‘5’]  с помощью метода **join** | print ( " ".join (A) ) # метод **join** соединяет в одну строку те строки, которые  # стоят в скобках в качестве параметра. Строки  # соединяются в данном примере через пробел – он указан  # в кавычках перед **join** |
| Вывести элементы числового массива:  A = [‘2’, ‘4’, ‘6’, ‘5’]  с помощью метода **join** | print ( " ".join (map (str, A)))  # метод **join** соединяет в одну строку только строки,  # поэтому элементы массива надо преобразовать в str.  # Сделаем это с помощью функции map. |