Домашняя работа от 08.02.2024.  
Выполнил Суганов Борис

Группа МИСТ 23-3-3

**Дополнение**:

При тестировании использовался декоратор для установки затраченного времени

Код:

import datetime  
  
  
def decorator(func):  
 def wrapper(\*args, \*\*kvargs):  
 startTime = datetime.datetime.now()  
 result = func(\*args, \*\*kvargs)  
 print("Время исполнения:", datetime.datetime.now() - startTime)  
 return result  
  
 return wrapper

**Задание 1.**

Определить значение и номер последнего отрицательного элемента массива

Программа:

@decorator

def task\_1(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 for i in range(len(array) - 1, -1, -1):  
 if array[i] < 0:  
 print(f'Результат: "{array[i]}" - значение, "{i}" - номер(индекс)')  
 break

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задание 2.**

Из элементов заданного массива сформировать два новых массива, включая в первый массив элементы исходного массива с четными индексами, во второй – с нечетными.

Код:

@decorator

def task\_2(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 length = len(array)  
 if length % 2 == 0:  
 first\_array = [None] \* (length // 2)  
 else:  
 first\_array = [None] \* (length // 2 + 1)  
 sec\_array = [None] \* (length // 2)  
  
 for i in range(len(array)):  
 index = i // 2  
 if i % 2 == 0:  
 first\_array[index] = array[i]  
 else:  
 sec\_array[index] = array[i]  
  
 print(f'Результат: \n{first\_array} - массив со значениями четных индексов')  
 print(f'{sec\_array} - массив со значениями нечетными индексов')

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задание 3.**

Найти сумму квадратов элементов, расположенных до первого отрицательного элемента массива.

Код:

@decorator

def task\_3(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 i = 0  
 sum\_array = 0  
 while array[i] >= 0:  
 sum\_array += array[i] \*\* 2  
 i += 1  
 print(f'Результат: {sum\_array}')

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

**Задание 4.**

Определить индексы элементов массива, меньших среднего. Результат получить в виде массива.

Код:

@decorator

def task\_4(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
  
 avg = sum(array) / len(array)  
 result = [i for i in range(len(array)) if array[i] < avg]  
 print(f'Результат: {result}')

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задание 5.**

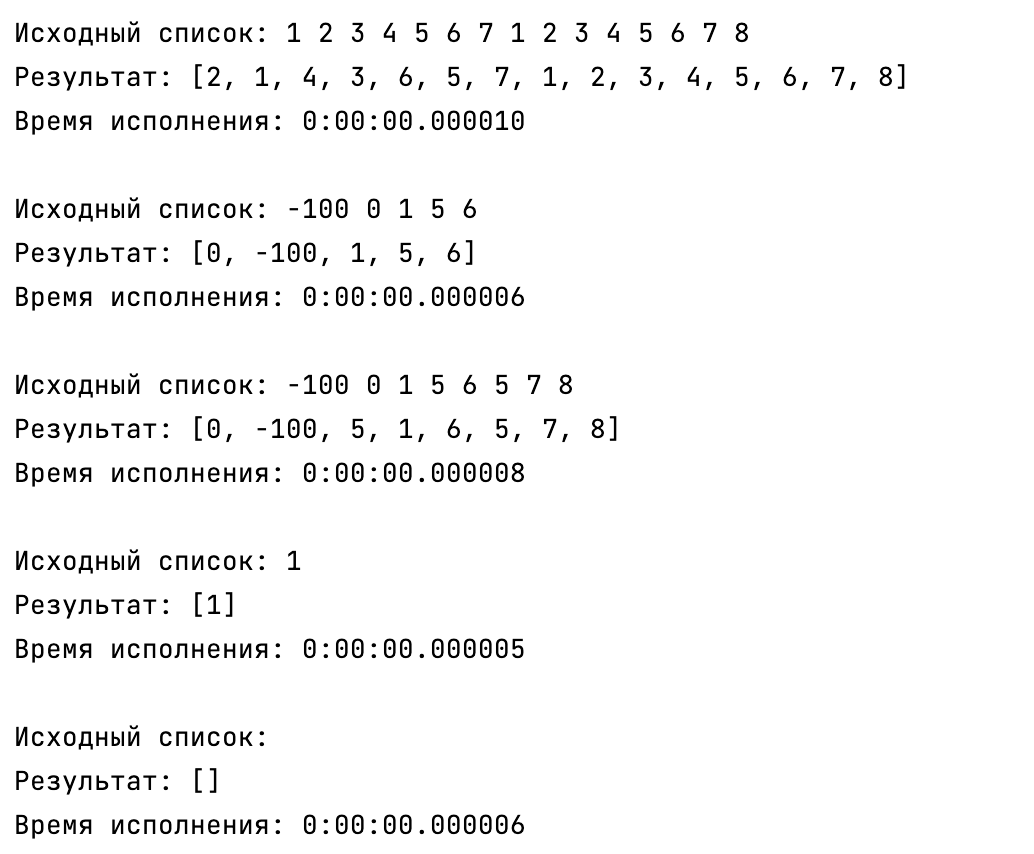
В одномерном массиве поменять местами соседние элементы (1-й со 2-м, 3-й с 4-м и т.д.), распложенные в первой половине массива.

Код:

@decorator

def task\_5(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 mid = len(array) / 2 - 2  
 i = 0  
 while i <= mid:  
 if i % 2 == 0:  
 array[i], array[i + 1] = array[i + 1], array[i]  
 i += 1  
 print(f'Результат: {array}')

Результаты:



**Задание 6.**

В массиве *А* найти максимальное количество следующих подряд упорядоченных по убыванию элементов.

Код:

@decorator

def task\_6(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 count = 1  
 if len(array) == 0:  
 count = 0  
 a = 0  
 b = 0  
 for i in range(1, len(array)):  
 if array[i] < array[i - 1]:  
 count += 1  
 b = i  
 else:  
 if count < b - a:  
 count = b - a  
 a = i  
 b = i  
  
 print(f'Результат: {count}')

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

**Задание 1\*.**

В массиве находятся целые числа от (в произвольном порядке), одно – один раз, остальные -  по 2 раза. Найти число, встречающееся 1 раз.

Код:

@decorator

def task\_1\_star(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 length = len(array)  
 sum = 0  
 sub\_array = [None] \* length  
 for i in range(len(array)):  
 if array[i] in sub\_array:  
 sum -= array[i]  
 else:  
 sub\_array[i] = array[i]  
 sum += array[i]  
 print(f'Результат: {sum}')

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

**Задание 2\*\*.**

В массиве находятся целые числа, больше половины из которых равны Х. Найти Х.

Код:

@decorator

def task\_2\_star(array):  
 print(f'Исходный список: {" ".join(str(el) for el in array)}')  
 answer = array[0]  
 max\_count = 1  
 count\_in\_row = 1  
 for i in range(len(array)):  
 if array[i] == array[i - 1]:  
 count\_in\_row += 1  
 if count\_in\_row > max\_count:  
 max\_count = count\_in\_row  
 answer = array[i]  
 else:  
 count\_in\_row = 1  
  
 print(f'Результат: {answer}')

Результаты:  
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задание 3\*\*.**

В массиве находятся целые числа, одно – один или 2 раза, остальные -  по 3 раза. Найти это число и сколько раз оно встречается.

Код:

@decorator

def task\_3\_star(arr):  
 print(f'Исходный список: {" ".join(str(el) for el in arr)}')  
 anwer\_count = len(arr) % 3  
 first\_array = [None] \* len(arr)  
 first\_sum = 0  
 sec\_array = [None] \* len(arr)  
 sec\_sum = 0  
 for i in range(len(arr)):  
 if arr[i] not in first\_array:  
 first\_array[i] = arr[i]  
 first\_sum += arr[i]  
 elif arr[i] not in sec\_array:  
 sec\_array[i] = arr[i]  
 else:  
 sec\_sum += arr[i]  
 print(f'Результат: "{first\_sum - sec\_sum}" - число встречается {anwer\_count} раз(а)')

Результаты:

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание