**ФГБОУ ВО “МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

**Лабораторная работа № 3**

Разветвляющиеся вычислительные процессы

**Задание 1 Вариант № 10**

по дисциплине:

Основы программирования

Выполнил:

студент 1 курса

группы 201-322

Кириллов Д.П

**МОСКВА 2020**

**Постановка задачи:**

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

* Минимальный по модулю элемент массива
* Найти сумму модулей элементов в массиве после первого нуля
* Преобразовать массив таким способом, чтобы первая половина элементов стояла на четных позициях, а вторая часть элементов на нечетных

**Теоретическая часть:**

Для решения задачи были использованы списки, встроенная функция abs(), циклы, условные операторы. Для организации массива – конечной последовательности однотипных величин был использован список, заполненный с помощью цикла for. Также была использована функция append, которая позволяет добавить элемент в конец списка.

**Описание программы:**

Программа написана на алгоритмическом языке Python 3.8.5, реализована в среде ОС Windows 10

**Описание алгоритма:**

1. Ввод с клавиатуры количества элементов
2. Ввод элементов массива
3. Вывод исходного массива
4. Нахождение с помощью функции minArr() минимальное значение списка по модулю
5. Вывод минимального значения
6. Нахождение суммы модулей элементов списка после первого нуля с помощью функции sumAfterZero()
7. Преобразование списка с помощью функции changeArr(), которая ставит первую часть элементов на четные позиции, а вторую на нечетные
8. Вывод преобразованного списка

**Листинг программы:**

arr = []

try:

arrayLength = int(input('Введите длину массива: '))

# Ввод элементов массива

print('Введите элементы массива')

for i in range(arrayLength):

arr.append(int(input('> ')))

# Функция возвращающая минимальное число в массиве

def minArr(arr):

min = abs(arr[0])

for i in range(arrayLength):

if (abs(arr[i]) < min):

min = arr[i]

return min

# Сумма модулей элементов после нуля

def sumAfterZero(arr):

firtZeroIndex = -1

sum = 0

# Находим индекс первого элемента равного нулю

for i in range(arrayLength):

if arr[i] == 0:

firtZeroIndex = i

break;

# Находим сумму модулей элементов расположенных после firtZeroIndex

for i in range(firtZeroIndex, arrayLength):

sum += abs(arr[i])

return sum

# Преобразовывает массив таким образом, что первая половина элементов

# массива стоят на четных позициях, а вторая на нечетных

def changeArr(arr):

newArr = []

firstHalf = -1

if (arrayLength % 2 == 1):

firstHalf = int(((arrayLength - 1)/2))

else:

firstHalf = int(arrayLength/2)

for i in range(firstHalf):

newArr.append(arr[i])

if (i != firstHalf - 1):

newArr.append(None)

for i in range(firstHalf, arrayLength):

if (i != firstHalf):

newArr.append(None)

newArr.append(arr[i])

return newArr

def changeArr(arr):

arrFirst = []

arrSecond = []

newArr = [0] \* arrayLength

indexFirstArr = 0

indexSecondArr = 1

if (arrayLength % 2 == 1):

firstHalf = int(((arrayLength - 1)/2))

else:

firstHalf = int(arrayLength/2)

for i in range(firstHalf):

arrFirst.append(arr[i])

for i in range(firstHalf, arrayLength):

arrSecond.append(arr[i])

for i in range(len(arrFirst)):

newArr[indexFirstArr] = arrFirst[i]

indexFirstArr += 2

for i in range(len(arrSecond)):

newArr[indexSecondArr] = arrSecond[i]

indexSecondArr += 2

return newArr

print('Исходный массив: {}'.format(arr))

print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

print('Минимальное по модулю число: {}'.format(minArr(arr)))

print('Сумма модулей элементов после нуля: {}'.format(sumAfterZero(arr)))

print('Преобразованный массив: {}'.format(changeArr(arr)))

except ValueError:

print('Введите корректное число.')

**Результат тестирования программы**

Исходный массив: [1, 2, 3, 0, -6, 0, 7, 8, 10, 23]

----------------------------------------

Минимальное по модулю число: 0

Сумма модулей элементов после нуля: 54

Преобразованный массив: [1, None, 2, None, 3, None, 0, None, -6, 0, None, 7, None, 8, None, 10, None, 23]

**Использованная литература**

pythonworld.ru Справочник для начинающих. Python 3