**Урок No7. Списки**

1.

# Задание No1

# В первой строке вводится число N. Далее в N строк вводится N чисел (1 ≤ N ≤

# 10000), по одному числу на строке. Все числа по модулю не превышают 10e5.

# Переверните массив чисел. Выведите N чисел - перевернутый массив.

#

# https://github.com/A-l-E-v/PySynergy/blob/main/U-7/reverse.py

#

print()

print('--- Перевёрнутый массив ---')

print()

N=int(input('Введите размерность N='))

n\_array = []

# заполняем массив

for i in range (N):

number = int(input ('Введите число: '))

n\_array.append (number)

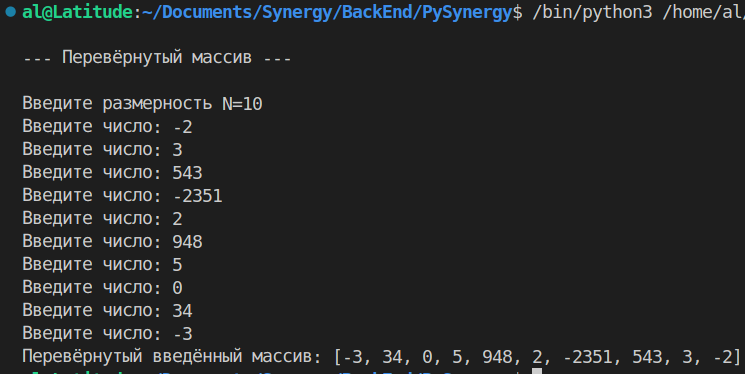
# переворачиваем массив в новый массив

n\_array.reverse()

# выводим новый массив

print('Перевёрнутый введённый массив:', n\_array)

Проверяю работу программы на 10 элементах:



2.

# Задание No2

# В первую строчку вводится число N (1 ≤ N ≤ 100 000). В следующую строку

# через пробел вводятся N чисел (1 ≤ Ai ≤ 10e9). Вам требуется написать метод,

# который получает на вход массив и изменяет его таким образом, чтобы на

# первом месте стоял последний элемент, на втором - первый, на третьем -

# второй и т. д. Выведите N чисел - измененный массив.

#

# https://github.com/A-l-E-v/PySynergy/blob/main/U-7/array.py

#

print()

print('--- Последний на первом ---')

print()

N=int(input('Введите размерность N='))

n\_array = []

# получаем исходный массив

n\_array = list(map (int, input('Вводите числа через пробел: ').split()))

#вставляем последний элемент на первое место копированием

n\_array.insert(0,n\_array[-1])

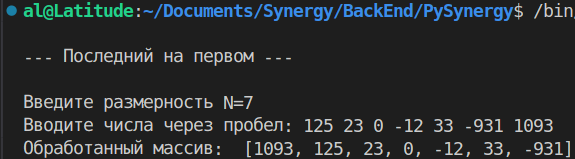
# удаляем последний перенесённый элемент

n\_array.pop()

# выводим результирующий массив

print ('Обработанный массив: ', n\_array)

Проверяю работу программы на 7 элементах:



3.

# Задание No3

# На берегу реки стояли n рыбаков, все они хотели перебраться на другой берег.

# Одна лодка может выдержать не более m килограмм, при этом в лодку

# помещается не более 2 человек. Определите, какое минимальное число лодок

# нужно, чтобы перевезти на другой берег всех рыбаков В первую строку

# вводится число m (1 ≤ m ≤ 10e6) - максимальная масса, которую может

# выдержать одна лодка. Во вторую строку вводится число n (1 ≤ n ≤ 100) -

# количество рыбаков. В следующие N строк вводится по одному числу Ai (1 ≤ Ai

# ≤ m) - вес каждого путешественника. Программа должна вывести одно число -

# минимальное количество лодок, необходимое для переправки всех рыбаков

# на противоположный берег.

#

# https://github.com/A-l-E-v/PySynergy/blob/main/U-7/boat.py

#

print()

print('--- Лодочная переправа ---')

print()

m = float(input ('Введите массу в килограммах, которую может выдержать одна лодка: '))

n = int (input('Введите количество рыбаков: '))

# массив весов рыбаков

a =[]

# по очереди вводим веса рыбаков

for i in range (n):

a.append(float (input('Введите вес '+str(i+1)+' рыбака в килограммах: ')))

if max(a) > m:

print('Масса рыбака превышает грузоподъёмность лодки!')

exit(0)

# сортируем массив весов рыбаков от самого тяжёлого до самого лёгкого

a.sort(reverse=True)

print('Веса рыбаков в порядке убывания: ', a)

# массив лодок c 2 или с 1 тяжёлым рыбаком

boats = []

# временный массив для обработки

temp = a

for x in range(len(temp)):

# если можно вдвоём

if temp[x] + min(temp) <= m:

# сажаем самого тяжёлого и самого лёгкого

boats += [[temp[x], min(temp)]]

# чтобы избежать дублей по минимальному весу, увеличиваем на грузоподъёмность лодки

temp[x] += m

temp[temp.index(min(temp))] += m

else:

if temp[x] > m:

continue

else:

# сажаем одного в лодку

boats += [[temp[x]]]

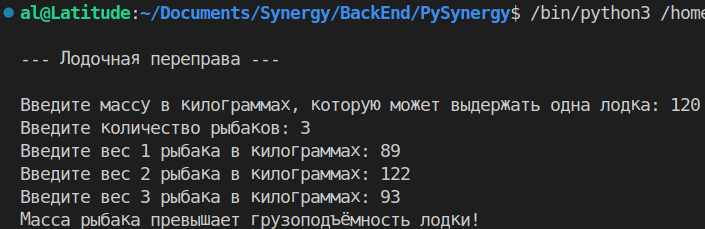
# рассадка рыбаков

print ('Рыбаки поплывут в парах или по одному ([a,a]):', boats)

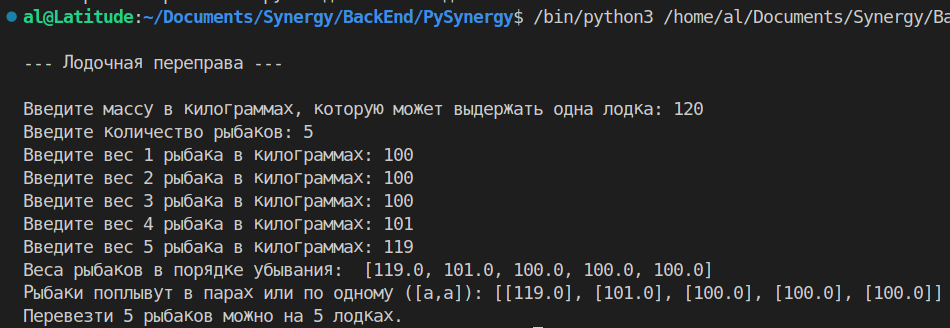
# количество лодок - длина массива b

print('Перевезти '+str(n)+' рыбаков можно на '+str(len(boats))+' лодках.')

Проверяю случай, когда переправа не возможна — хотя бы один из рыбаков тяжелее для лодки:



Проверяю всех очень тяжёлых — каждому потребуется по лодке:



Проверяю обычный случай, переправа возможно и по 1 и по 2 рыбакам:

