Практическое занятие № 6

**Тема:** составление программ со списками в IDE

PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы,

основные принципы составления программ, приобрести навыки составление

программ со списками в IDE PyCharm Community.

**Постановка задачи:**

1. Дан целочисленный список A размера 10. Вывести порядковый номер последнего из

тех его элементов A K, которые удовлетворяют двойному неравенству A 1 < A K < A 10 .

Если таких элементов нет, то вывести 0.

2. Дано число R и список размера N. Найти два различных элемента списка, сумма

которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания

их индексов (определение наиболее близких чисел - то есть такой элемент A K для

которого величина |A K - R| является минимальной).

3. Дан список размера N, все элементы которого, кроме первого, упорядочены по

возрастанию. Сделать список упорядоченным, переместив первый элемент на новую

позицию.

**Текст программы:**

#1. Дан целочисленный список A размера 10. Вывести порядковый номер последнего из  
#тех его элементов AK, которые удовлетворяют двойному неравенству A1 < AK < A10.  
#Если таких элементов нет, то вывести 0.  
import random  
A=[]  
for i in range(10):  
 A.append(random.randint(0, 20))  
print(A)  
k = 0  
for i in range(10):  
 if (A[0] < A[i]) and (A[i] < A[9]):  
 k = i +1  
print(k)

[3, 6, 3, 19, 15, 11, 4, 20, 1, 12]

7

Process finished with exit code 0

#2. Дано число R и список размера N. Найти два различных элемента списка, сумма  
#которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания  
#их индексов (определение наиболее близких чисел - то есть такой элемент AK, для  
#которого величина |AK - R| является минимальной).  
import random  
def closest\_numbers(R, lst):  
 index1 = 0  
 index2 = 0  
 min\_diff = abs(lst[0] + lst[1] - R)  
 for i in range(len(lst)):  
 for j in range(i+1, len(lst)):  
 diff = abs(lst[i] + lst[j] - R)  
 if diff < min\_diff:  
 index1 = i  
 index2 = j  
 min\_diff = diff  
 return index1, index2  
  
R = int(input("Введите число R: "))  
while type(R) != int:  
 try:  
 number\_x = int(R)  
 except ValueError:  
 print("Неправильно ввели!")  
 R = int(input("Введите число R: "))  
lst = []  
N = int(input("Введите размер списка: "))  
for i in range(N):  
 lst.append(random.randint(0, 20))  
print(lst)  
result = closest\_numbers(R, lst)  
  
print("Индекс пары наиболее близких элементов:", result)

Введите число R: 15

Введите размер списка: 10

[16, 17, 20, 1, 5, 7, 8, 3, 9, 19]

Индекс пары наиболее близких элементов: (5, 6)

Process finished with exit code 0

#3. Дан список размера N, все элементы которого, кроме первого, упорядочены по  
#возрастанию. Сделать список упорядоченным, переместив первый элемент на новую  
#позицию.  
  
my\_list =[8,1,2,3,5,7,9]  
elem = my\_list.pop(0)  
new\_list = []  
for i in my\_list:  
 if i>elem:  
 new\_list.append(elem)  
 new\_list.append(i)  
print(new\_list)

**[1, 2, 3, 5, 7, 8, 9]**

**Process finished with exit code 0**

**Вывод:** в процессе выполнения практического занятия выработала навыки составления программ структуры с ветвлениями в IDE PyCharm Community. Были использованы языковые конструкции if, функции, цикл for, проверка условий и математические операции.

**Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.**