**Практическое занятие № 6**

**Тема:** Составление программ линейной структуры в IDE PyCharm Community.

**Цель:** выработка первичных навыков работы с IDE PyCharm Community, составление программ линейной структуры.

**Постановка задачи 6.1.**

Дан список ненулевых целых чисел размера N . Проверить образуют ли его элементы геометрическую прогрессию. Если образуют, то вывести знаменатель прогрессии, если нет - вывести 0

**Тип алгоритма:** отсутствует

**Текст программы:**

# вариант 24  
# Дан список ненулевых целых чисел размера N .  
# Проверить образуют ли его элементы геометрическую прогрессию.  
# Если образуют, то вывести знаменатель прогрессии, если нет - вывести 0  
def check\_geometric\_progression(arr):  
 if len(arr) < 2:  
 return 0  
  
 ratio = arr[1] / arr[0]  
  
 for i in range(1, len(arr)):  
 if arr[i] / arr[i - 1] != ratio:  
 return 0  
  
 return ratio  
  
  
# Пример использования:  
arr = [2, 4, 8, 16, 32]  
result = check\_geometric\_progression(arr)  
if result:  
 print(f"Элементы образуют геометрическую прогрессию с знаменателем {result}")  
else:  
 print("Элементы не образуют геометрическую прогрессию")

**Протокол работы программы:**

**Пример:** arr = [2, 4, 8, 16, 32]  
result = check\_geometric\_progression(arr)  
if result:  
 print(f"Элементы образуют геометрическую прогрессию с знаменателем {result}")  
else:  
 print("Элементы не образуют геометрическую прогрессию")

**Результат:** **Элементы образуют геометрическую прогрессию с знаменателем 2.0**

**Process finished with exit code 0**

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ линейной структуры в IDE PyCharm Community были использованы. Были использованы языковые конструкции int,input (),print – вывод полученного значения, списки

Выполнены разработка кода, отладка, тестирование, оптимизация кода.

Готовые программные коды выложены на GitHub.

**Постановка задачи 6.2.**

Дан целочисленный список A размера N . Переписать в новый целочисленный список B того же размера вначале все элементы исходного списка с чётными номерами, а затем - с нечётными : A2, A4,....., A1, A3, A5,....... . Условный оператор не использовать

**Тип алгоритма:** отсутствует

**Текст программы:**

# 24 вариант  
# Дан целочисленный список A размера N .  
# Переписать в новый целочисленный список B того же размера вначале все элементы исходного списка с чётными номерами  
# , а затем - с нечётными : A2, A4,....., A1, A3, A5,....... . Условный оператор не использовать  
A = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
B = A[1::2] + A[0::2]  
  
print(B)

**Протокол работы программы:**

**Результат: [2, 4, 6, 8, 1, 3, 5, 7, 9]**

**Process finished with exit code 0**

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ линейной структуры в IDE PyCharm Community были использованы. Были использованы языковые конструкции int,input (),print – вывод полученного значения, списки

Выполнены разработка кода, отладка, тестирование, оптимизация кода.

Готовые программные коды выложены на GitHub.

**Постановка задачи 6.3.**

Дано множество A из N точек (точки заданы своими координатами x, y) Найти пару различных точек этого множества с максимальным расстоянием между ними и само это расстояние ( точки выводятся в том же порядке, в котором они перечислены при задании множества A). Расстояние R между точками с координатами (x1, y1) и (x2, y2) вычисляется по формуле R=√(x2-x1) ²+ (y2-y1)² . Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй - для хранения ординат

**Тип алгоритма:** отсутствует

**Текст программы:**

import math  
  
  
def max\_distance(points):  
 max\_dist = 0  
 max\_dist\_points = (0, 0), (0, 0)  
  
 for i in range(len(points)):  
 x1, y1 = points[i]  
 for j in range(i + 1, len(points)):  
 x2, y2 = points[j]  
 dist = math.sqrt((x2 - x1) \*\* 2 + (y2 - y1) \*\* 2)  
 if dist > max\_dist:  
 max\_dist = dist  
 max\_dist\_points = (x1, y1), (x2, y2)  
  
 return max\_dist\_points, max\_dist  
  
  
# пример использования функции  
points = [(1, 2), (3, 4), (5, 6), (7, 8)]  
print(max\_distance(points))

**Протокол работы программы:**

**Пример:**points = [(1, 2), (3, 4), (5, 6), (7, 8)]  
print(max\_distance(points))

**Результат: (((1, 2), (7, 8)), 8.48528137423857)**

**Process finished with exit code 0**

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ линейной структуры в IDE PyCharm Community были использованы. Были использованы языковые конструкции int,input (),print – вывод полученного значения, списки, оператор return

Выполнены разработка кода, отладка, тестирование, оптимизация кода.

Готовые программные коды выложены на GitHub.