

Практическое занятие № 6

Тема: Составление программ со списками в IDE PyCharm Community.

Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

Постановка задачи 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.

Текст программы:

```
A = [] # Ввод данных
N = int(input('Размер списка: '))
t = N
k = 1
u = 0
q = 0
s = N - 2
while t: # Создание списка
    A.append(int(input('Значение списка: ')))
    t -= 1
p = A[k] / A[u] # Решение
while s:
    k += 1
    u += 1
    if A[k] / A[u] == p:
        q += 1
```

```
s -= 1  
  
if q == N - 2: # Проверка условия и вывод результата  
    print(int(p))  
  
else:  
    print(0)
```

Протокол работы программы:

Размер списка: 4

Значение списка: 1

Значение списка: 2

Значение списка: 4

Значение списка: 8

2

Process finished with exit code 0

Постановка задачи 2.

Дан целочисленный список A размера N. Переписать в новый целочисленный список B того же размера в начале все элементы исходного списка с чётными номерами, а затем - с нечётными: A2, A4, A6,..., A1, A3, A5,... . Условный оператор не использовать.

Текст программы:

```
A = [] # Ввод данных  
  
B = []  
  
a = 0  
  
b = 1  
  
N = int(input('Размер списка: '))  
  
t = N  
  
q = N - 1
```

```
while t: # Создание списка

    A.append(int(input('Значение списка: ')))

    t -= 1

print(A)

while b <= q: # Решение

    B.append(A[b])

    b += 2

while a <= q:

    B.append(A[a])

    a += 2

print(B) # Вывод данных
```

Протокол программы:

Размер списка: 5

Значение списка: 12

Значение списка: 34

Значение списка: 56

Значение списка: 78

Значение списка: 90

[12, 34, 56, 78, 90]

[34, 78, 12, 56, 90]

Process finished with exit code 0

Постановка задачи 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:
 $R = \sqrt{((x2-x1)**2) + ((y2-y1)**2)}$. Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй - для хранения ординат.

Текст программы:

```
import math
```

```
A = [] # Ввод данных
```

```
B = []
```

```
C = []
```

```
N = int(input('Кол-во точек: '))
```

```
t = N
```

```
n = N - 2
```

```
g = n
```

```
a = 0
```

```
b = 1
```

```
R3 = 0
```

```
v = 0
```

```
e1 = 0
```

```
e2 = 0
```

```
while t: # Создание списка
```

```
    A.append(input('Точка: '))
```

```
    B.append(int(input('Значение по абсциссе: ')))
```

```
    C.append(int(input('Значение по ординате: ')))
```

```
    t -= 1
```

```
while n: # Решение
```

while g:

R1 = math.sqrt(((B[b]-B[a])**2)+((C[b]-C[a])**2))

p = b

b += 1

R2 = math.sqrt(((B[b] - B[a]) ** 2) + ((C[b] - C[a]) ** 2))

if R1 >= R2:

if R3 < R1:

R3 = R1

e1 = a

e2 = p

elif R2 >= R1:

if R3 < R2:

R3 = R2

e1 = a

e2 = b

g -= 1

b = 1

v += 1

a += 1

b += v

n -= 1

g = n

print('Точки: ', A[e1], A[e2]) # Вывод результата

print('Расстояние: ', R3)

Протокол работы программы:

Кол-во точек: 7

Точка: A

Значение по абсциссе: -5

Значение по ординате: 12

Точка: B

Значение по абсциссе: 12

Значение по ординате: 6

Точка: C

Значение по абсциссе: 4

Значение по ординате: 8

Точка: D

Значение по абсциссе: -12

Значение по ординате: 16

Точка: E

Значение по абсциссе: 2

Значение по ординате: 18

Точка: F

Значение по абсциссе: 3

Значение по ординате: 6

Точка: S

Значение по абсциссе: -1

Значение по ординате: 9

Точки: B D

Расстояние: 26.0

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции while, if. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.