

Мета: навчитися розрізняти основні види циклічних операторів, використовувати їх властивості при складанні відповідних синтаксичних конструкцій; закріпити правила застосування керуючих операторів break, continue; перетворювати один вид циклічного оператора в інший, складати програми з використанням циклічних алгоритмів на мові Python в інтерактивному режимі IDLE

Обладнання: ОС Windows, Microsoft Office2013, середовище програмування
IDLE

Хід роботи

1 Використовуючи головне меню ОС Windows запусив інтерактивний режим інтерпретатора Python – середовище програмування IDLE : Пуск→Всі програми→Python 3.7→IDLE Python 3.7 (32 або 64 - bit)

2 Створив в своєму каталозі папку ЛР8 для збереження наступних завдань.

3 Використовуючи функцію range() вивів на екран наступні значення

Print(list(range(-3,4)))	<i>-3 -2 -1 0 1 2 3</i> >>>
Print(list(range(17,3,-2)))	<i>17 15 13 11 9 7 5 3</i> >>>
Out = 1 For I in range(1,8): Out *= 2 Print(out, end=' ')	<i>2 4 8 16 32 64 128</i> >>>
out = 243 print(out,end=' ') for i in range(1,6): out /= 3 print(out,end=' ')	<i>243 81 27 9 3 1</i> >>>

					ДТЗЕ.121РПЗ00ТЕХ0800ЛР							
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка програм з використанням циклічних алгоритмів			Літ.	Арк.	Аркуші		
Розроб.		Мітленко Д.А.								1	7	
Перевір.		Саприкіна І.Г.						Група ПЗ-19-1/9				
Реценз.												
Н. Контр.		Саприкіна І.Г.										
Затверд.		Саприкіна І.Г.										

4 За допомогою команди File→NewFile (Ctrl+N) створbd нові текстові документи з відповідними назвами ЛР_8(№ завдання) для розробки наступних кодів програми

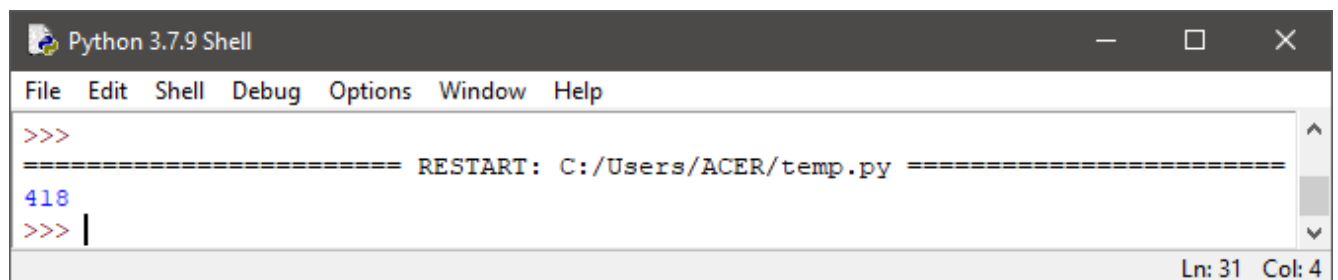
5 Виконав індивідуальні завдання виданого варіанту (варіант 1)

5.1 Складіть програму, що обчислює суму всіх парних чисел в діапазоні від 13 до 31.

Код програми:

```
suma = 0
for i in range(13,32):
    if (i%1) == 0:
        suma += i
print(suma)
```

Результат зображено на скріншоті 1



Скріншот 1

5.2 Складіть програму, що обчислює добуток чисел такої колекції: 2,5,1,7,3,9,6.

Код програми:

```
col = {2,5,1,7,3,9,6}
dob = 1
for i in col:
    dob *= i
print(dob)
```

Результат зображено на скріншоті 2

		Мітленко Д.А.			ДТЗЕ.121РПЗ00ТЕХ0800ЛР	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/ACER/temp.py =====
11340
>>> |
```

Ln: 34 Col: 4

Скріншот 2

5.3 Задано рядок символів: “мінімальна кількість слів”. Знайдіть його довжину, кількість букви «і» та виведіть його на екран в стовпчик

Код програми:

```
s = "мінімальна кількість слів"
print("len =",len(s))
print("кількість і =",len(s.split("і"))-1)
```

Результат зображено на скріншоті 3

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/ACER/temp.py =====
len = 25
кількість і = 5
>>> |
```

Ln: 38 Col: 4

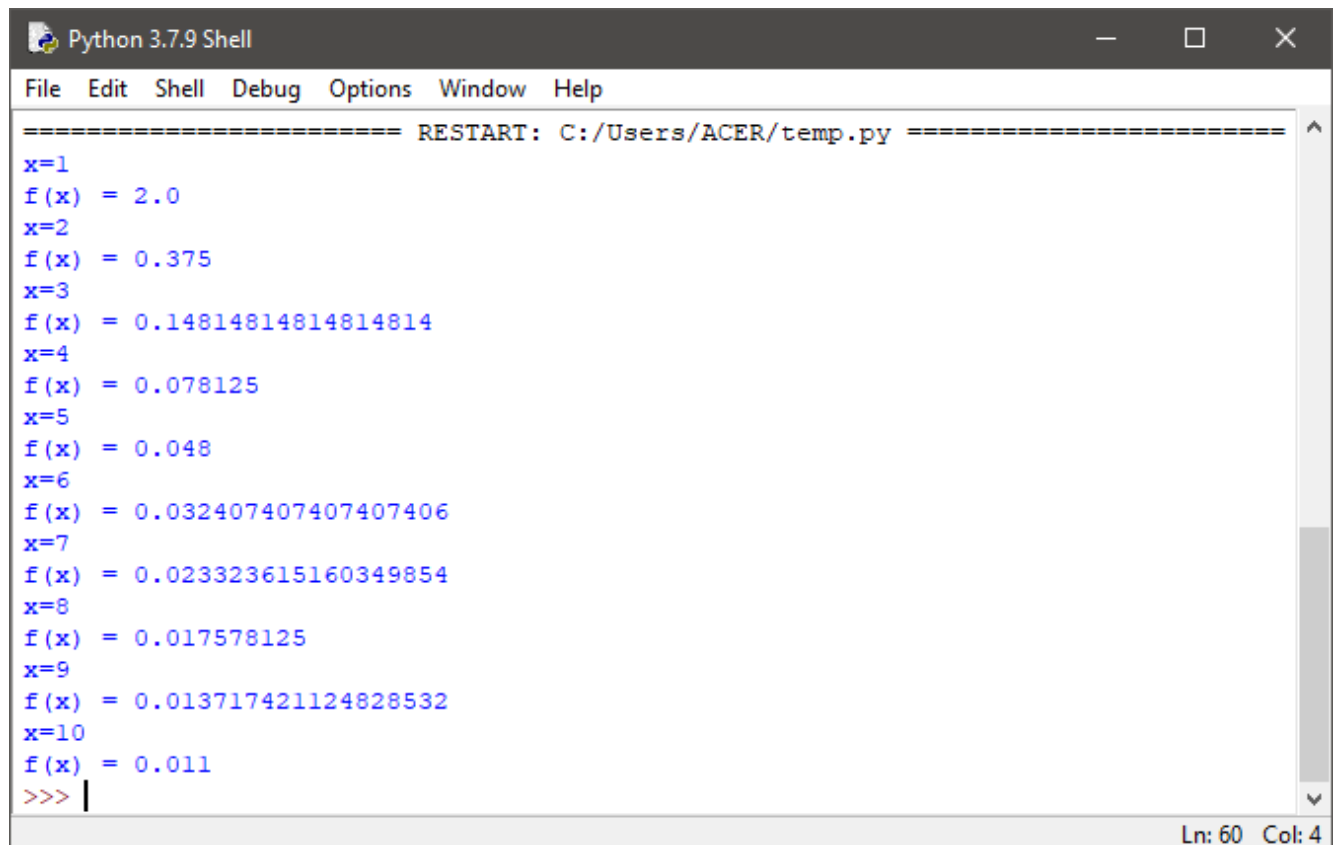
Скріншот 3

5.4 Обчислити значення функцій $f_1(x) = \frac{x+1}{x^3}$ для значень аргументу $x = 1, 2, 3, \dots, 10$

Код програми:

```
f = lambda x: (x + 1)/(x**3)
for i in range(1,11):
    print("x=%i\nf(x) ="%i,f(i))
```

Результат зображений на скріншоті 4



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/ACER/temp.py =====
x=1
f(x) = 2.0
x=2
f(x) = 0.375
x=3
f(x) = 0.14814814814814814
x=4
f(x) = 0.078125
x=5
f(x) = 0.048
x=6
f(x) = 0.032407407407407406
x=7
f(x) = 0.023323615160349854
x=8
f(x) = 0.017578125
x=9
f(x) = 0.013717421124828532
x=10
f(x) = 0.011
>>> |
```

Ln: 60 Col: 4

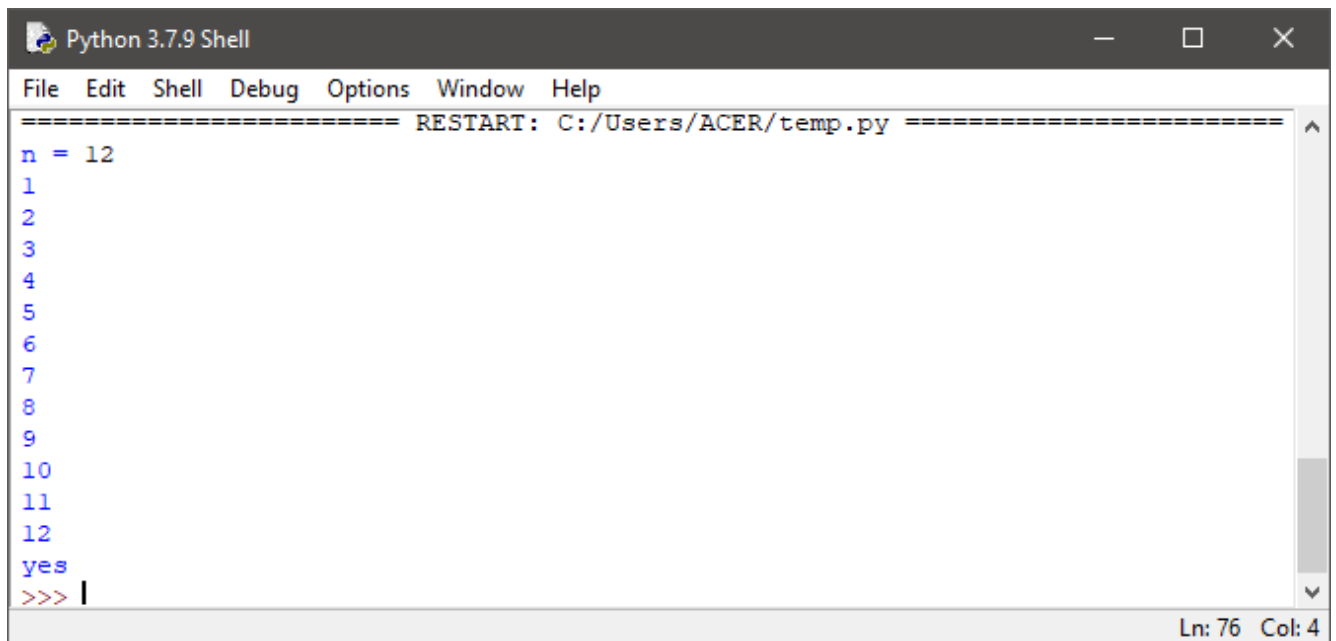
Скріншот 4

5.5 Напишіть програму, яка доводить або перевіряє, що для безлічі натуральних чисел виконується рівність: $1 + 2 + \dots + n = n(n + 1) / 2$, де n - будь-яке натуральне число.

Код програми:

```
n = int(input("n = "))
suma = 0
for i in range(1,n+1):
    print(i)
    suma += i
if suma == (n*(n+1)/2): print("yes")
else: print("no")
```

Результат зображений на скріншоті 5



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/ACER/temp.py =====
n = 12
1
2
3
4
5
6
7
8
9
10
11
12
yes
>>> |
```

Ln: 76 Col: 4

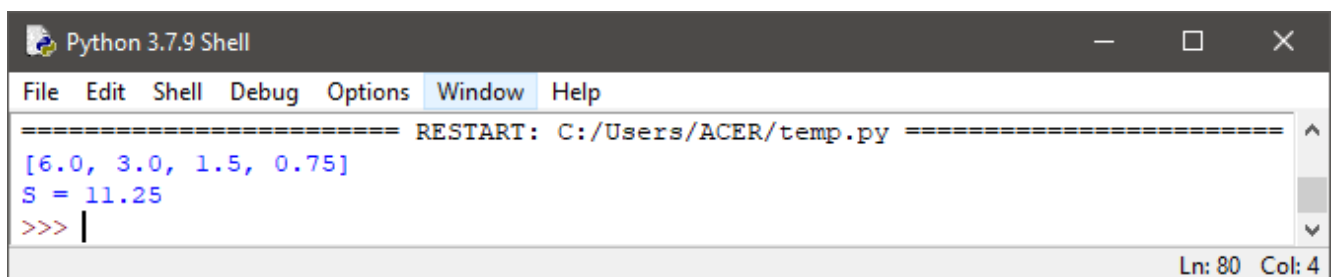
Скріншот 5

5.6 Перший член геометричної прогресії дорівнює 6, а її знаменник – 0.5. Розробіть код обчислення значень членів прогресії, більших за 0.6 і визначення номера останнього члена прогресії, що підсумовується.

Код програми:

```
f = 6
q = 0.5
prog = []
for i in range(1,5):
    if (f * q ** (i - 1)) >= 0.6:
        prog.append(f * q ** (i - 1))
print(prog)
print("S =", f * (1-q**len(prog))/(1-q))
```

Результат зображено на скріншоті 6



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Users/ACER/temp.py =====
[6.0, 3.0, 1.5, 0.75]
S = 11.25
>>> |
```

Ln: 80 Col: 4

Скріншот 6

Відповіді на контрольні запитання

1 Який алгоритм називають циклічним? Які основні види циклів є в програмуванні?

Циклічний алгоритм – це алгоритм, який повторює свої дії вказану кількість разів. Є два основні види: цикл з умовою, цикл з параметром.

2 Які основні етапи розрізняють в циклах? Що таке тіло циклу? Які інструкції воно може містити? Що таке ітератор? Для чого він використовується?

Основні етапи: підготовчий, тіло циклу, керівний. Тіло циклу – частина коду, яку інтерпретатор буде повторювати до тих пір поки умова повторення циклу буде залишатися істинною. Ітератор - змінна, що надає послідовний доступ до елементів колекції та навігацію по ним.

3 Дайте означення поняттю «цикл по колекції»? Що таке колекція? Наведіть приклади різних колекцій.

Цикл по колекції – це структура керування, що здійснює повторення деякої послідовності інструкцій, для всіх елементів колекції. Колекція - це об'єкт (структура), що містить скінченний набір елементів, до яких реалізовано доступ, наприклад: {1, 2, 3, 4, 5, 6, 7, 8, 9}, “рядок символів”, ['а', 'б', 'в', 'г', 'д'];

4 Яка функція генерує колекцію в мові Python? Чи можна використовувати генератор колекції без аргументів? Поясніть чому.

range() - спеціальна функція яка повертає послідовність чисел в заданому діапазоні. Генератор колекції не можна використовувати без аргументів, бо ми задаємо аргументами діапазон чисел.

5 Що означає аргумент step? Які значення по замовчуванню містять аргументи генератора колекцій start та end?

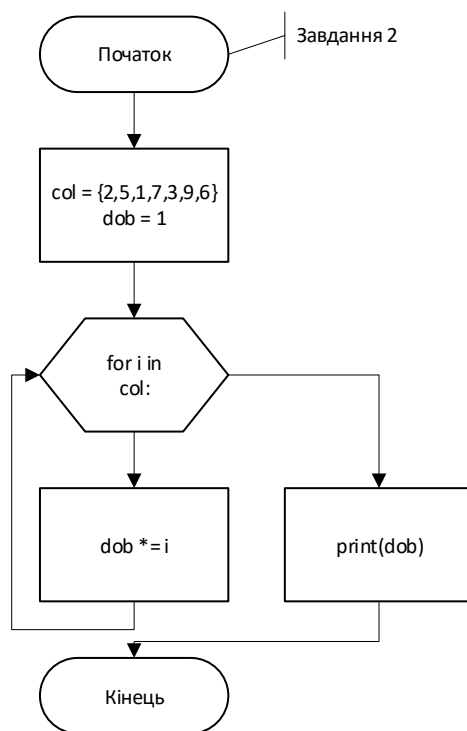
step – крок зміни ітератора. Початкове значення start – 0, а end не має початкового значення, його надає користувач.

6 В якому випадку ітератор циклу використовують як лічильник циклу?

Коли в тілі циклу потрібно знати номер ітерації

		Мітленко Д.А.			ДТЗЕ.121РПЗ00ТЕХ0800ЛР	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

7 Опишіть за допомогою блок-схеми розв'язок будь-якого завдання зі свого варіанту та опишіть які елементи були використані в цій схемі. Поясніть як відбувається цей циклічний процес



В цій блок-схемі були використані такий основний елемент як цикл. Цей циклічний процес відбувається так: поки ітератор знаходиться в колекції чисел то присвоюємо і одночасно помножаємо змінну добутку на значення ітератора.

Висновок: я навчився розрізняти основні види циклічних операторів, використовувати їх властивості при складанні відповідних синтаксичних конструкцій; закріпити правила застосування керуючих операторів break, continue; перетворювати один вид циклічного оператора в інший, складати програми з використанням циклічних алгоритмів на мові Python в інтерактивному режимі IDLE