# Лабораторна робота 3 Циклічні процеси в Python

**Мета роботи:** Ознайомитись з операторами циклу в Python. Побудувати блок-схему алгоритму, розробити програму з використанням циклів, реалізувати її та оцінити результат її роботи.

## Рейтинг лабораторної роботи №3

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Вересень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 4 |  |

## Теоретичний матеріал

В Python існують два типи циклічних виразів:

* 1. цикл while
  2. цикл for

### Цикл while в Python

Інструкція while в Python повторює вказаний блок коду до тих пір, поки вказаний в циклі логічний вираз буде залишатися істинним.

**while логічний вираз:**

**команда 1**

**команда 2**

**...**

**команда n**

Після ключового слова **while** вказується умовний вираз, і поки цей вислів повертає значення **True**, буде виконуватися блок інструкцій, який йде далі. Всі інструкції, які відносяться до циклу **while**, розташовуються на наступних рядках і повинні мати відступ від початку рядка (4 пробіли).

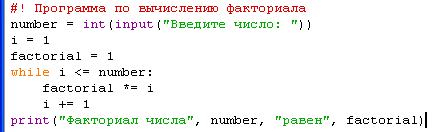


Рис.1. Приклад циклу while для обчислення факторіалу числа

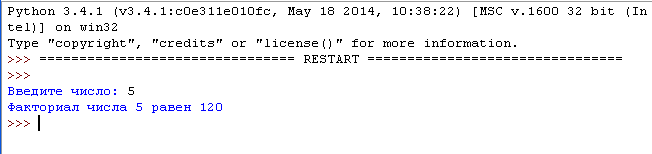


Рис.2. Результат обчислення факторіалу числа

### Цикл for

Оператор **for** виконує вказаний набір інструкцій задану кількість разів, яка визначається кількістю елементів в наборі (наприклад, в рядках або списках).

Синтаксис циклу for:

**for int in range ():**

**команда 1**

**команда 2**

**...**

**команда n**

Змінній **int** присвоюється значення першого елемента функції **range (),** після чого виконуються команди.

Потім змінній **int** присвоюється наступне по порядку значення і так далі до тих пір, поки не будуть перебрані всі елементи функції **range ().**

**Приклад.** Знайти суму n елементів наступного ряду чисел: 1 -0.5 0.25 -0.125 ... n. Кількість елементів (n) вводиться з клавіатури. Вивести на екран кожен член ряду і його суму. Вирішити задачу використовуючи циклічну конструкцію **for**.

**Рішення:**

1. В даному випадку ряд чисел складається з елементів, де кожен наступний менше попереднього в два рази по модулю і має зворотний знак. Значить, щоб отримати наступний елемент, треба попередній розділити на -2.
2. На початку будь-якій змінній треба присвоїти значення першого елемента ряду (в даному випадку це 1).
3. Далі в циклі додавати її значення до змінної, в якій накопичується сума, після чого присвоїти їй значення наступного елемента ряду, розділивши поточне значення на -2.
4. Цикл повинен виконуватися n раз.

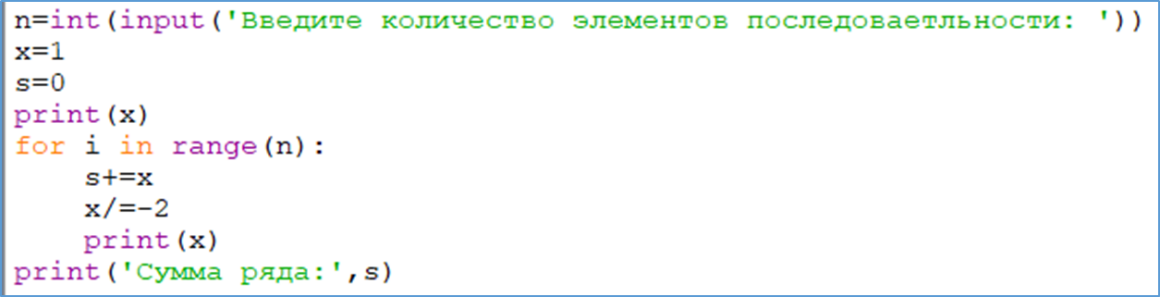


Рис.3. Приклад циклу for для обчислення суми послідовності чисел

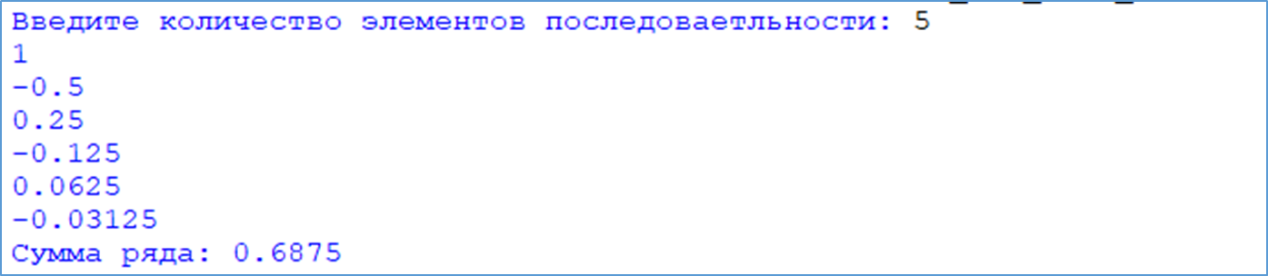


Рис.4. Результат обчислення суми послідовності чисел

### 3. Оператори break і continue

При роботі з циклами використовуються оператори **break і continue.**

Оператор **break** призначений для дострокового переривання роботи циклу **while** або **for.**

Оператор **continue** запускає цикл заново, при цьому код, розташований після цього оператора, не виконується.

Блок **else** дозволяє перевірити чи виконався цикл повністю. Якщо ключове слово **break** не було викликане, то буде виконаний блок **else**.

Приклад. Перевірити, чи задане число є простим.

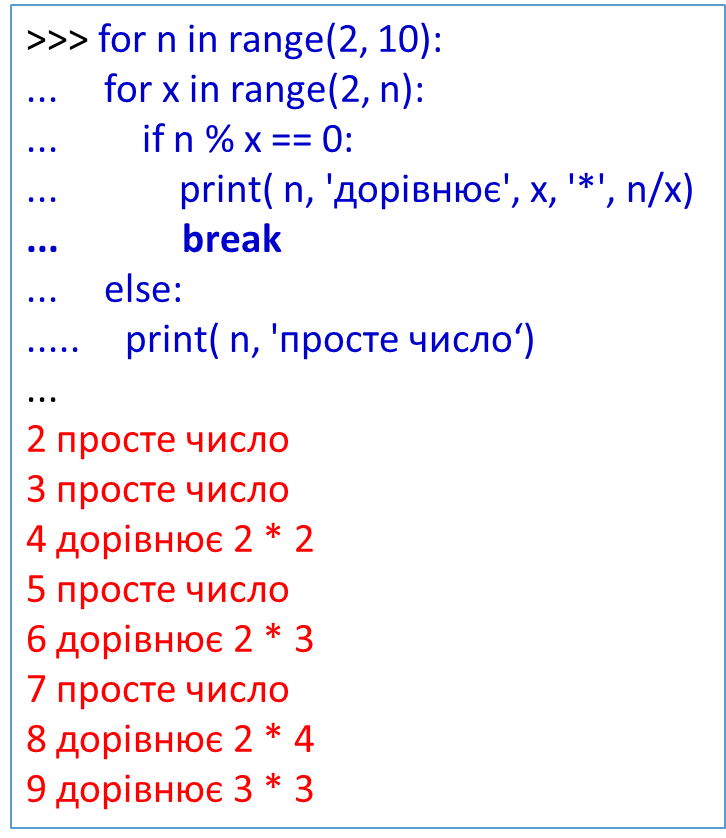


Рис.5. Код програми перевірки числа на простоту та результати його роботи.

## Завдання для самостійної роботи (за варіантами)

Обчислити суму ряду, елементи якого розраховуються з використанням операцій піднесення у степінь, розрахунку факторіалу числа, операцій множення та ділення.

Для розрахунку першої суми з параметром від 1 до 5 використати цикл **for**. Для розрахунку внутрішньої суми з параметром від 0 до нескінченності використати цикл **while**.

Для визначення нескінченності потрібно задати точність розрахунку поточного элементу ряду. Підсумовування припиняється, якщо досягнута задана користувачем точність розрахунку. Перевірити знаменник на рівність нулю для запобігання ділення на нуль.

Результат подати у вигляді таблиці із чотирма колонками, що визначають параметри зовнішньої та внутрішньої сум, значення члена ряду та накопичувану суму (рис. 6)

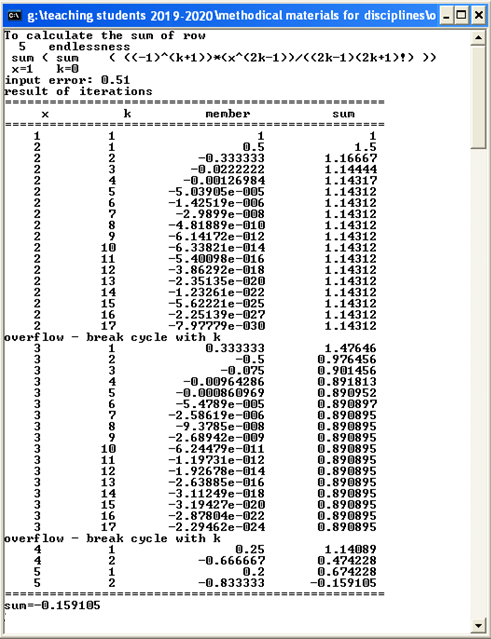
****

Рис. 6. Лістинг результату роботи програми обчислення суми ряду

### Варіанти завдань

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var1.gif | 12. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var12.gif |
| 2. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var2.gif | 13. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var13.gif |
| 3. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var3.gif | 14. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var14.gif |
| 4. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var4.gif | 15. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var15.gif |
| 5. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var5.gif | 16. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var16.gif |
| 6. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var6.gif | 17. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var17.gif |
| 7. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var7.gif | 18. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var18.gif |
| 8. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var8.gif | 19. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var19.gif |
| 9. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var9.gif | 20. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var20.gif |
| 10. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var10.gif | 21. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var21.gif |
| 11. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var11.gif | 22. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var22.gif |

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.