# Лабораторна робота 5 Рекурсивні функції в Python

**Мета роботи:** Ознайомитись з технологією використання рекурсивних функцій в Python. Побудувати блок-схему алгоритму, розробити програму з використанням рекурентних співвідношень для реалізації рекурсивних процесів.

## Рейтинг лабораторної роботи №5

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Жовтень |
| 2 | Захист роботи | 0,5 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 0,5 |
| Разом за роботу | | 3 |  |

## Теоретичний матеріал

Означення називається рекурсивним, якщо воно задає елементи множини за допомогою інших елементів цієї самої множини. Об'єкти, що задані рекурсивним означенням, також називаються рекурсивними. Під терміном "рекурсивно задана функція" розуміється така функція, значення якої для даноrо apгумента обчислюється за допомогою значень для попередніх apгументів. При кожній конкретній реалізації процес обчислення такої функції задається *рекурентною послідовністю*.

Рекурсія — це такий спосіб організації обчислювального процесу, за якого функція звертається сама до себе. Такі звернення називаються *рекурсивними викликами,* а функція, що містить рекурсивні виклики, — *рекурсивною*.

Перед тим, як записати рекурсивну функцію будь-якою мовою програмування, необхідно записати рекурентне співвідношення, що визначає метод обчислення функцій. Рекурентне співвідношення повинне містити як мінімум дві умови:

* умову продовження рекурсії (крок рекурсії);
* умову закінчення рекурсії.

Розглянемо функцію , де аргумент  є цілим додатним числом, таким, що  або . Потрібно обчислити значення членів послідовності  Рекурсивне завдання послідовності  як послідовності значень функції  включає такі два етапи.

1. Функція  задається безпосередньо у вигляді числових значень для деякої конечної множини початкових значень аргументу .
2. Задається метод або формула, які дозволяють, знаючи усі значення функції при , обчислити її значення при , тобто знайти .

В результаті отримуємо peкурентні співвідношення, що описують рекурсивно задану функцію , рівносильну послідовності  .

Отриманий запис цих двох етапів



називається рекурентним співвідношенням.

Рекурсію реалізовують за допомогою виклику функції самої себе. При цьому в тілі функції спочатку слід перевіряти умову продовження рекурсії. Якщо вона істинна, то виходити з функції. Інакше здійснювати рекурсивний крок.

В рекурсивних підпрограмах можна виділити два процеси: *рекурсивне занурення* підпрограми у себе, що відбувається доти, доки параметр не сягне граничного значення, та *рекурсивне повернення* з підпрограми, що відбувається доти, доки параметр не сягне початкового значення. Рекурсія реалізує нескінченний цикл, отже застосовувати її слід тільки у випадках, коли без неї неможливо обійтися.

Розглянемо рекурсивну функцію, що обчислює *факторіал числа*. Значення факторіалу визначаються рекурентним співвідношенням:



Програмний код, що реалізує рекурсивні обчислення, такий:

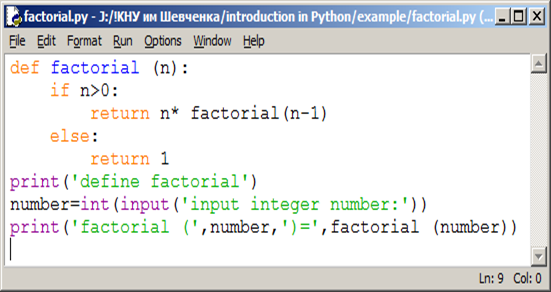


Рис.1. Код обчислення факторіалу числа

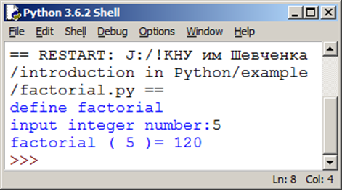


Рис.2. Результат обчислення факторіалу числа

## Завдання для самостійної роботи (за варіантами)

Для виконання завдань використовувати рекурсивні виклики функцій. Для оцінки ефективності рекурсії визначити її глибину.

### Варіанти завдань

1. Знайти суму цифр натурального числа *n* та значення глибини рекурсії, використовуючи рекурентне означення функції *f(n)*:



*Підказка*. Умова продовження рекурсії: сума цифр числа дорівнює останній цифрі плюс сума цифр числа без останньої цифри (числа, що ділиться без остачі на 10). Умова закінчення рекурсії: якщо число дорівнює 0, то сума його цифр дорівнює 0.

1. Знайти кількість одиниць в двійковому представленні числа *n* та значення глибини рекурсії*,* використовуючи рекурентне означення функції *f(n)* (& - операція побітового логічного множення):

****

1. Знайти значення біноміального коефіцієнта та глибини рекурсії при заданих *n, k* за формулою:

 = ,

використовуючи рекурентне співвідношення:



1. Знайти значення функції Аккермана *A(m, n),* використовуючи рекурентне співвідношення:



Визначити глибину рекурсії.

1. Визначити рекурсивну функцію та глибину рекурсії в процесі обчислення степеня дійсного числа *х* з цілим показником *n* ()*,* згідно з рекурентним співвідношенням:



1. Коефіцієнти розкладання бінома (*a+b*)*i*, тобто біноміальні коефіцієнти, утворюють *і*-й рядок трикутника Паскаля. Кожне число в трикутнику, крім перших трьох, є сумою чисел, розташованих над ним у попередньому рядку. Число в *i*-му рядку (*i*= 0, 1, 2, …) на *j-*му місці (*j*= 0, 1, …, *i*), задається формулою . «Верхівка» трикутника має наступний вигляд.



Надрукувати перші  рядків трикутника Паскаля, задавши значення *n* з клавіатури, використавши рекурсивне визначення його елементів.

1. Задані натуральні числа а, c, m. Визначити рекурсивну функцію та глибину рекурсії для обчислення  за формулою:

,

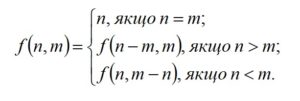
де  - залишок від ділення  на 10.

1. По заданим *b, p, m* обчислити значення виразу . Обчислення виконати, використовуючи алгоритм, що базується на двійковому розкладанні показника степені p:



Операція mod визначає остачу від ділення цілих чисел (ділення за модулем). Визначити глибину рекурсії.

1. Визначити найбільший спільний дільник двох цілих чисел за формулою Евкліда та розрахувати глибину рекурсії:

[](https://www.bestprog.net/wp-content/uploads/2019/01/05_02_02_06_07_01u.jpg)

1. Дано натуральне число *n*> 1. Перевірте, чи є воно простим. Програма повинна вивести слово YES, якщо число просте і NO, якщо число складене.

*Підказка*. потрібно зробити ще один параметр рекурсії: дільник числа, і саме за цим параметром і робити рекурсію.

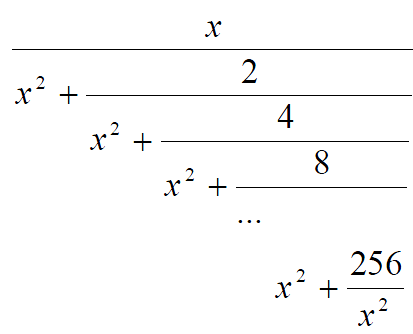
1. Дано натуральне число *n*>1. Вивести всі прості дільники цього числа в порядку неспадання з урахуванням кратності. Використати рекурсію та розрахувати її глибину.
2. Дано натуральне число *N*. Обчислити суму його цифр. Використати рекурсію та розрахувати її глибину
3. Визначити рекурсивну функцію NOD(*a, b*) цілого типу, яка знаходить найбільший спільний дільник (НСД) двох натуральних чисел *a* і *b*, використовуючи алгоритм Евкліда: НСД (*a, b*) = НСД (*b, a* mod *b*), якщо b≠0, НСД (a, 0) = a. За допомогою цієї функції знайти НСД (*a, b*), НСД (*a, c*), НСД (*a, d*), якщо задані числа *a,b,c,d.* Визначити глибину рекурсії. Операція mod визначає остачу від ділення двох цілих чисел.
4. Створити рекурсивну функцію для перекладу числа з десяткової системи числення в двійкову та з двійкової системи числення в десяткову. Визначити глибину рекурсії.
5. Дані два числа *a і b*. Визначити, скільки існує послідовностей з *a* нулів і *b* одиниць, в яких ніякі два нуля не стоять поруч. Оригінал умови за URL: <https://habr.com/ru/post/275813/>. Визначити глибину рекурсії.
6. . Описати рекурсивну функцію RootK (*X, K, N*), що знаходить наближене значення кореня *K*-го ступеня з числа *X* за формулою:



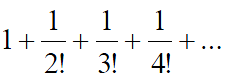
де *Yn* позначає RootK (*X, K, N*) при фіксованих *X і K*. Параметри функції:

X (> 0) - дійсне число, k > 1 і n > 0 - цілі. За допомогою функції RootK знайти для даного числа X наближені значення його кореня K-го ступеня при різних значеннях *n.* Оригінал умови за URL: <http://coolcode.ru/rekursiya-gruppa-recur/>

1. Дано натуральне число *n*. Обчислити суму ряду, елементи якого відповідають виразу: 1 × 2 + 2 × 3 + 3 × 4 +…+ (*n* –1) × *n*. Використати рекурсію та обчислити її глибину.
2. Обчислити ланцюговий дріб, задавши значення *х* з клавіатури та використавши рекурсію. Обчислити глибину рекурсії.



1. Знайти суму ряду, використавши рекурсивні функції обчислення факторіалу цілого числа та суми ряду. Визначити глибину рекурсії



1. Обчислити значення виразу



Для цього використати рекурсію при *n*=5, 10 (*n* – кількість членів послідовності). Розрахувати глибину рекурсії.

1. Обчислити значення *x =√ a* , використовуючи формулу , в якості початкового наближення використовувати значення *x*0 = (1+*a*)/2.
2. Обчислити добуток двох цілих додатних чисел p = a× b⋅ за наступним рекурентним співвідношенням:

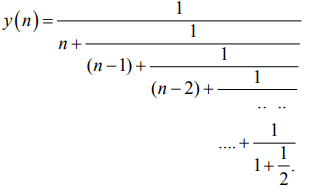


Використати рекурсію та розрахувати її глибину.

1. Обчислити значення полінома ступеня *n* за формулою

Використати рекурсію та розрахувати її глибину.

1. Обчислити ланцюговий дріб, задавши значення *n* з клавіатури та використавши рекурсію. Обчислити глибину рекурсії для заданого *n*



1. Обчислити значення виразу *m*!/(*m*+*n*)! при введених з клавіатури значеннях *m* i *n.* Використати рекурсивну функцію для обчислення факторіалу та розрахувати глибину рекурсії.

## 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.