MATH LIBRARY

ДОКУМЕНТАЦІЯ ДО РОЗРОБЛЕНОЇ БІБЛІОТЕКИ

ЗМІСТ

ПРО БІБЛІОТЕКУ

Код бібліотеки містить у собі десять функцій: *power*, *square_root*, *cube_root*, *factorial*, *combinations*, *placing*, *pascal_triangle*, *sum_geometric_progression*, *sum_infinity_geometric_progression* і *sum_arithmetic_progression*.
Блок-схема до кожної з них міститься на сторінках з їх описом.

Програма реалізована англійською мовою.

Для того щоб бібліотеку можна було використати у програмі, її необхідно імпортувати. Нижче наведений один із можливих варіантів імпорту у програму, яка написана на мові програмування Python:

```
import ctypes
from sys import platform
libc = ctypes.CDLL("library.dll")
```

Після імпорту можна викликати функції і використовувати їх у програмі.

POWER

(double x, double n)

ПАРАМЕТРИ:

x – число, яке необхідно піднести до степеня;
n – степінь.

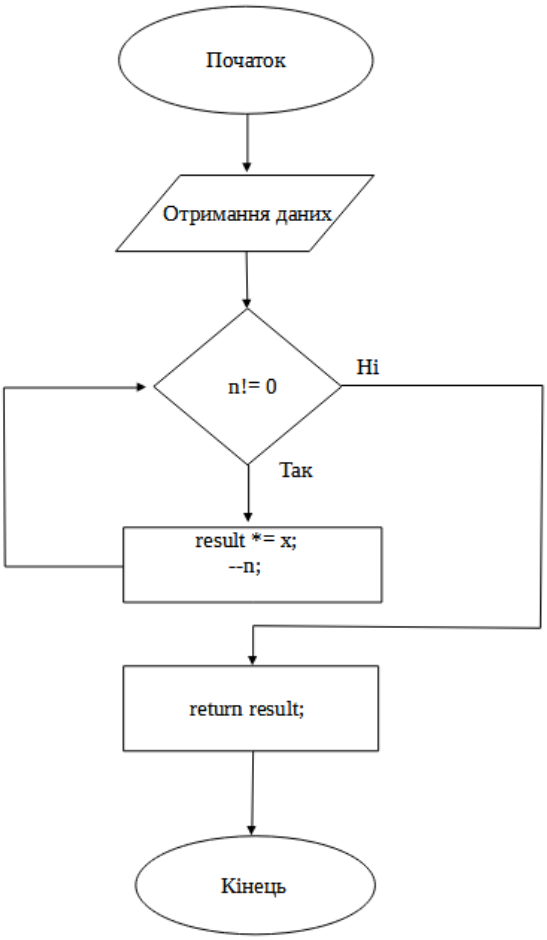
ОПИС

Спочатку вводиться змінна result, яка за замовчуванням дорівнює одиниці. Після цього вводиться цикл while (n != 0), який домножує число x на result n разів.
Після виконання циклу, функція повертає значення result.

ВИХІД:

double

БЛОК-СХЕМА ФУНКЦІЇ



1) SQUARE_ROOT (double n)
2) CUBE_ROOT

ПАРАМЕТРИ:

n – число, з якого будемо добувати корінь.

ОПИС

1) Для обчислення вводиться число і.
За допомогою циклу for ми перевіряємо число, яке може бути результатом. Тобто, шукаємо таке число, добуток якого його самого на себе буде дорівнювати нашому числу n.

Для точності дробової частини введено змінну precision, яка дорівнює 0.000001.

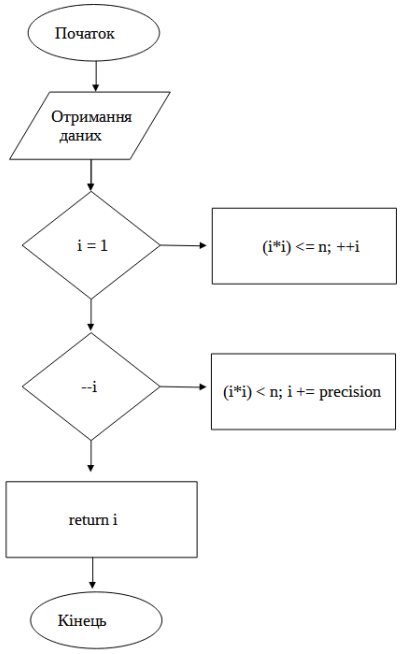
2) Для обчислення знову вводиться число і.
За допомогою циклу for ми перевіряємо число, яке може бути результатом. Тобто, шукаємо таке число, добуток якого його самого на себе і ще раз помножений на це число буде дорівнювати нашому числу n.

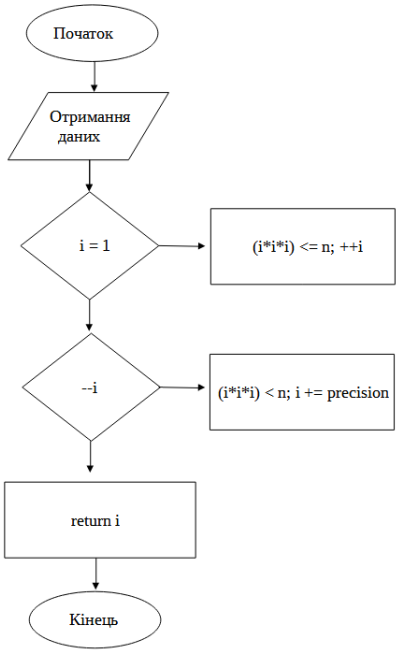
Для точності дробової частини введено змінну precision, яка дорівнює 0.000001.

ВИХІД:

1) double
2) double

БЛОК-СХЕМИ ФУНКЦІЙ





FACTORIAL (int n)

ПАРАМЕТРИ:

n – число, факторіал якого потрібно знайти.

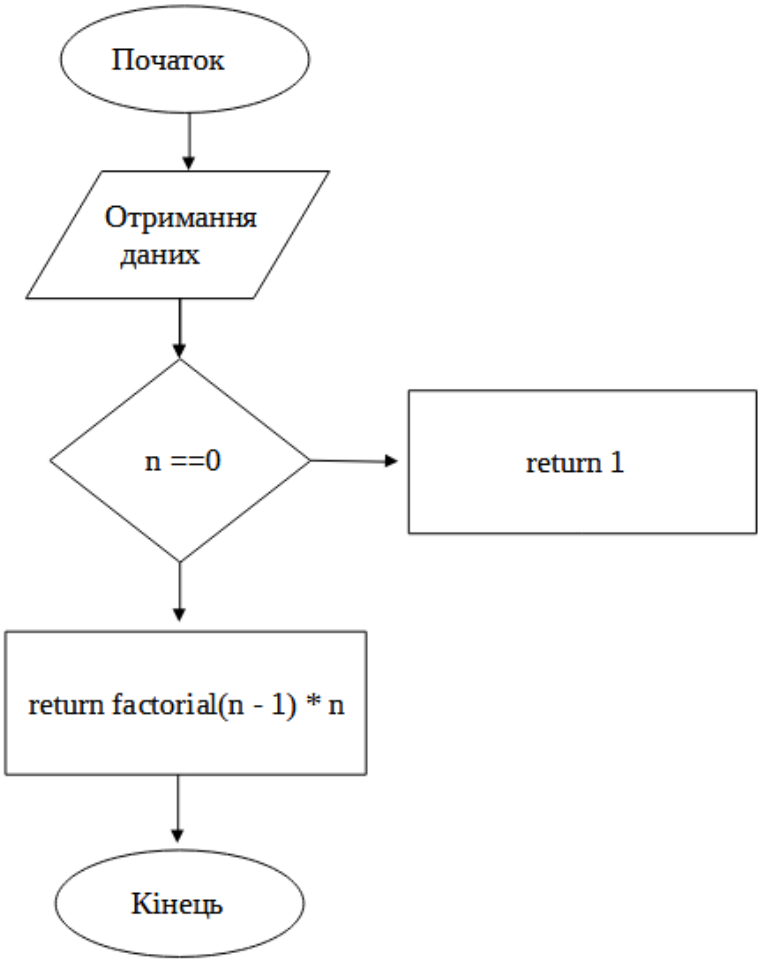
ОПИС

Функція є рекурсивною. Спочатку перевіряється чи є число n нулем. Якщо так, то функція повертає значення один. Якщо ні — продовжує свою роботу для n-1. Рекурсія буде відбуватися до того часу, допоки не буде виконана умова виходу з неї: n==0.

ВИХІД:

int

БЛОК-СХЕМА ФУНКЦІЇ



COMBINATIONS

(int n, int k)

ПАРАМЕТРИ:

n – кількість елементів множини;
k – сполуки множини.

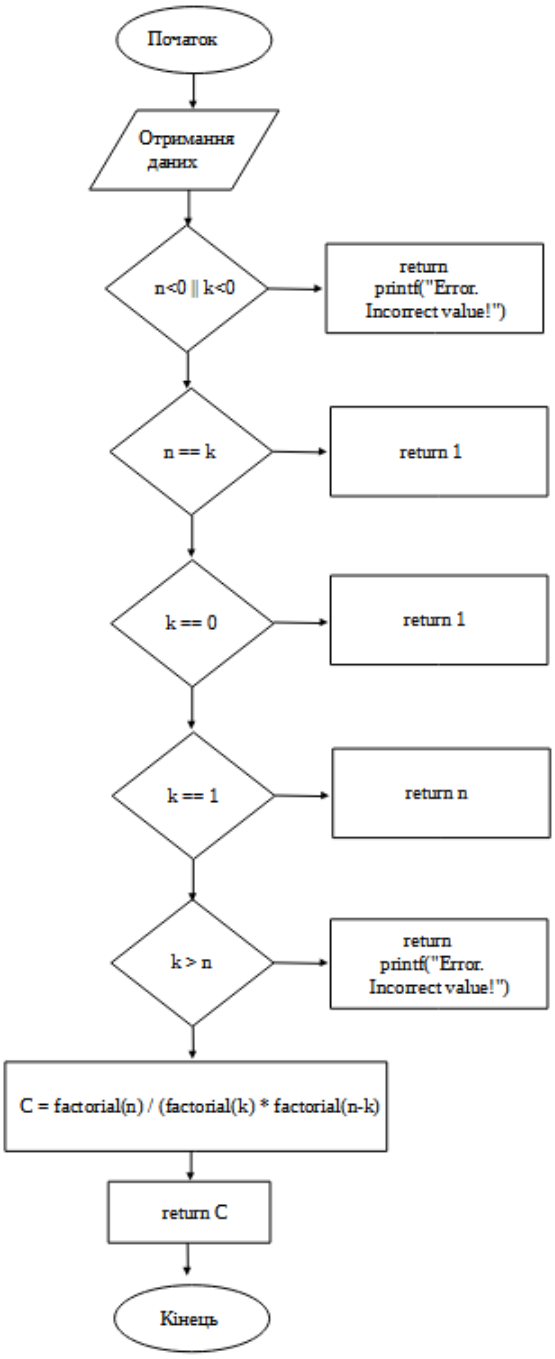
ОПИС

Спочатку функція перевіряє чи є n і k від'ємними числами. Якщо так, то повертає повідомлення про некоректний ввід. Далі перевіряється наступні умови: якщо n = k або k = 0, то функція повертає одиницю(за властивостями комбінації). Якщо k = 1, то функція повертає n (також, відповідно, за властивістю комбінації). Якщо k>n, то функція повертає повідомлення про некоректний ввід.

У випадку, якщо ці умови не були виконані, за формулою
C= n! / (k!(n-k)!) обраховується комбінація.

ВИХІД:

int



БЛОК-СХЕМА ФУНКЦІЇ

PLACING

(int n, int k)

ПАРАМЕТРИ:

n – кількість елементів множини;
k – сполуки множини.

ОПИС

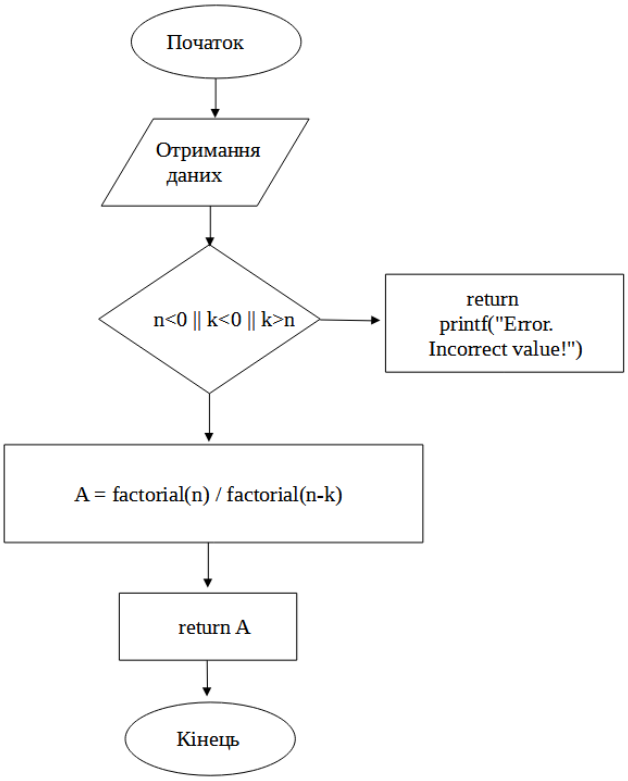
Спочатку функція перевіряє чи є n і k від'ємними числами. Якщо так, то повертає повідомлення про некоректний ввід. Також перевіряється чи k>n. У випадку позитивного результату функція повертає повідомлення про помилку.

Якщо умови не були виконані, то далі, за формулою A= n! / (n-k)! , обраховується розміщення.

ВИХІД:

int

БЛОК-СХЕМА ФУНКЦІЇ



PASCAL_TRIANGLE (int n)

ПАРАМЕТРИ:

n – показник степеня для біноміальних коефіцієнтів.

ОПИС

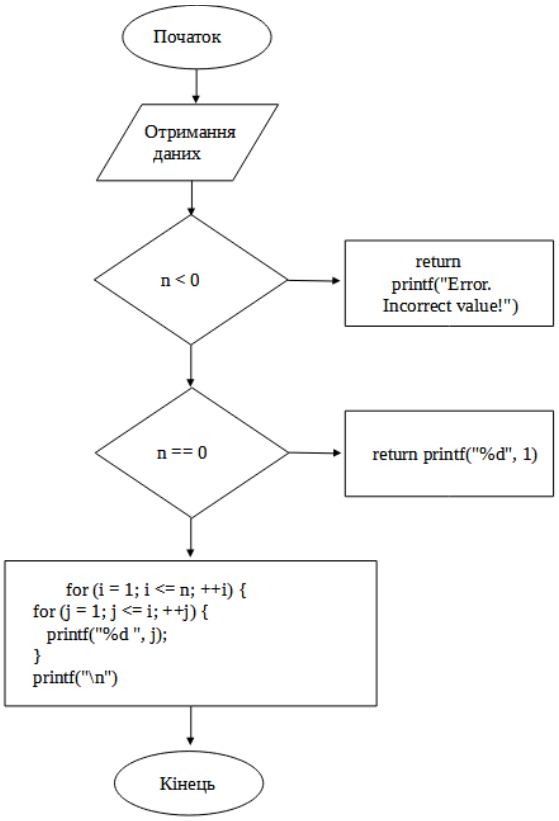
Спочатку функція перевіряє коректність вводу: чи є n<0. Якщо так, повертає повідомлення про помилку.

У випадку, коли n = 1, функція має повернути одиницю. Інакше, кожне число в кожному ряді одержуємо, додавши два числа, розміщені вгорі (зліва і справа). Якщо зліва або справа немає числа, підставляємо нуль на його місце.

ВИХІД:

int

БЛОК-СХЕМА ФУНКЦІЇ



SUM_ARITHMETIC_PROGRESSION (float a_1, float d, int n)

ПАРАМЕТРИ:

a_1 – перший елемент послідовності;
d – різниця прогресії;
n – кількість елементів у послідовності.

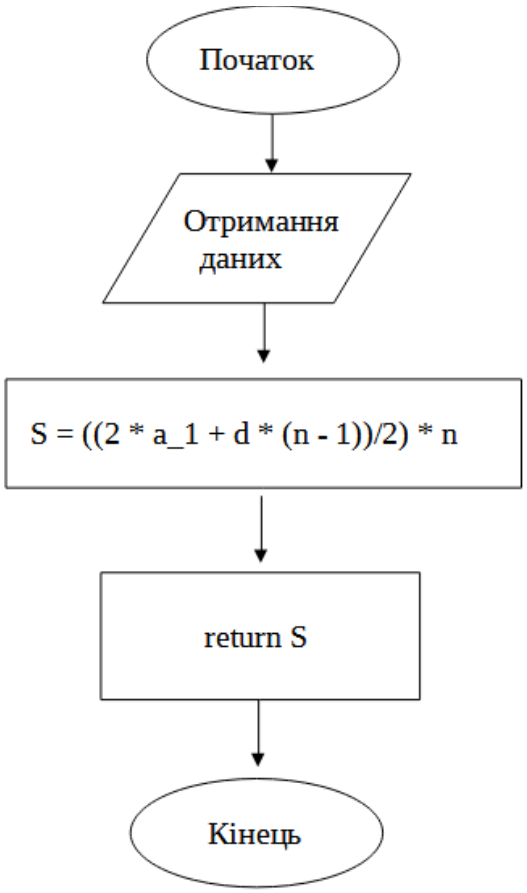
ОПИС

Функція реалізована досить просто: за допомогою однієї формули.
S = ((2a_1+d(n-1)) / 2) * n. Було використано цю формулу, через те, що вона є найпоширенішою.

ВИХІД:

float

БЛОК-СХЕМА ФУНКЦІЇ



SUM_GEOMETRIC_PROGRESSION (float b_1, float b_n, float q)

ПАРАМЕТРИ:

b_1 – перший елемент послідовності;
b_n – n-ий елемент послідовності;
q – знаменник прогресії.

ОПИС

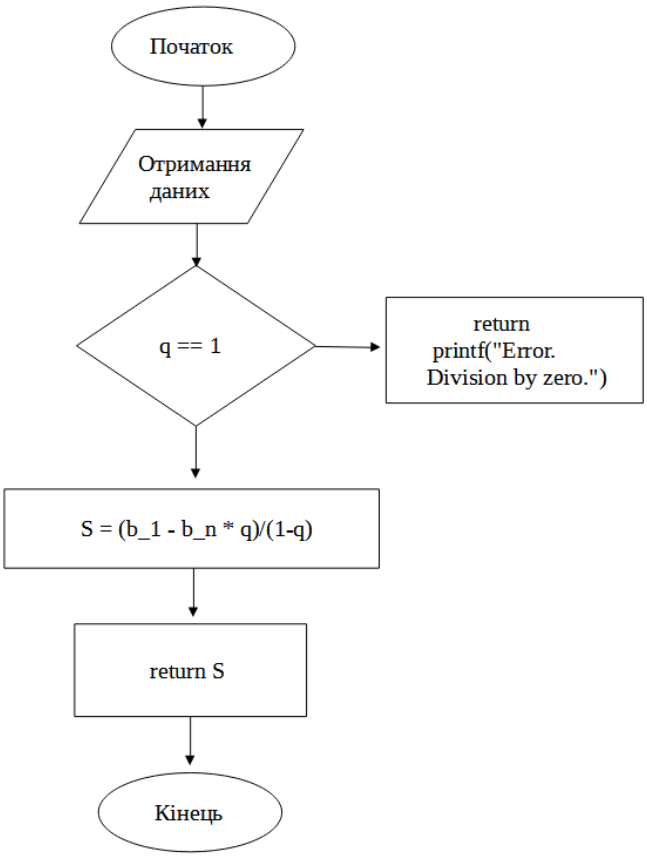
Ця функція також реалізована за допомогою однієї формули.
S = ((b_1-b_n q)) / (1-q). Цю формулу також було використано через її поширеність.

У випадку, якщо q = 1, буде повернено помилку.

ВИХІД:

float

БЛОК-СХЕМА ФУНКЦІЇ



SUM_INFINITY_GEOMETRIC_PROGRESSION (float b_1, float q)

ПАРАМЕТРИ:

b_1 – перший елемент послідовності;
q – знаменник прогресії.

ОПИС

У випадку, коли q = 1, функція поверне повідомлення про помилку.
Інакше, суму нескінченно спадної геометричної прогресії буде обраховано за формулою S = b_1 / (1-q).

ВИХІД:

float

БЛОК-СХЕМА ФУНКЦІЇ