# Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

# Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни

«Алгоритми і структури даних

»

Варіант 23

Виконав студент Панченко Сергій Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202

## Лабораторна робота 6

**Мета:** - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

**Постановка задачі**: . Написати рекурсивну функцію для обчислення суми цифр та кількості цифр натурального числа A. ( 23 варіант )

**Математична модель**:

**Складемо таблицю імен змінних**

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| n | string | Натуральне число | Проміжне значення |
| sum | Int | Сума цифр | Результат |
| count | Int | Кількість цифр | Результат |
| c | Int | Проміжна кількість цифр | Проміжне значення |
| S | Int | Проміжна  сума цифр | Проміжне значення |

Утворимо функцію def calc() та будемо викликати її рекурсивно. Перетворимо натуральне число у строку, та будемо виклакати calc() , поки довжина не стане 1, передаючи всі елементи, окрім першого. Далі значення calc() повертаємо кортежем в змінні c та s, додаємо їх до count та відповідно, повертаємо значення.

**Псевдокод:**

Крок 1: Визначимо основні дії

Крок 2: Деталізуємо calc()

**Крок 1:**

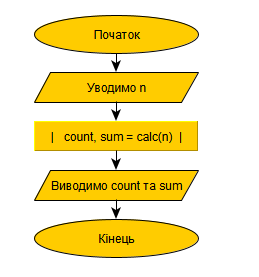
Початок

Уводимо n

count, sum := calc(n)

Виводимо sum та count

Кінець



Крок 2:

Початок

Уводимо n

count, sum := calc(n)

Виводимо sum та count

Кінець

Підпрограма calc(n):

Початок

count :=0

Якщо len(n) == 1:

sum :=int(n)

count :=1

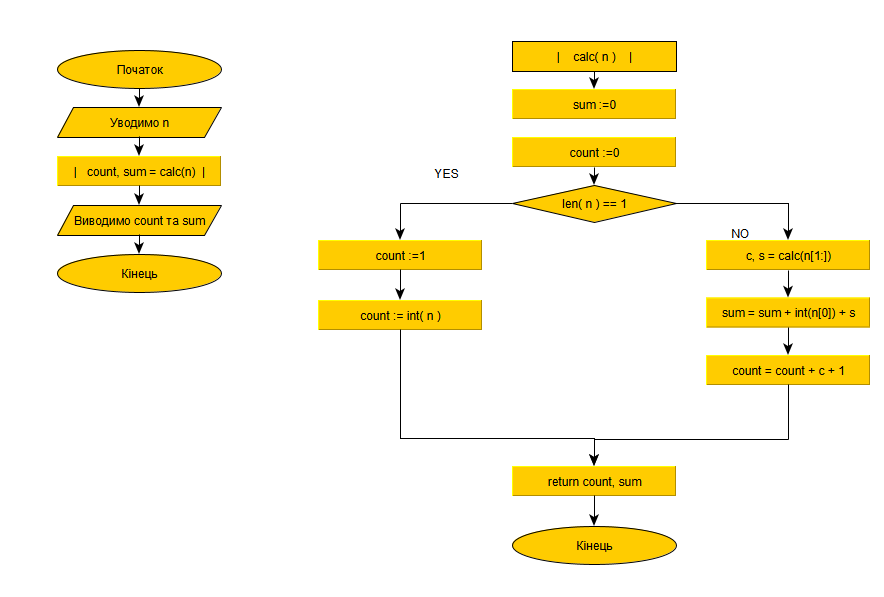
Інакше:

c, s = calc(n[1:])

sum = sum + int(n[0]) + s

count = count + c + 1

return count, sum



КОД:

def calc(n):

sum = 0

count = 0

if len(n) == 1:

count = 1

sum = int(n)

else:

c, s = calc(n[1:])

sum = sum + int(n[0]) + s

count = count + c + 1

return count, sum

count, sum = calc(input('Enter integer > 0: '))

print(f'Count: {count}\tSum: {sum}')

Перевірка:

Створимо таблицю для перевірки:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вкладеність** | **n** | **count** | **sum** | **c** | s |
| **1** | **245** | **-** | **-** | **-** | **-** |
| **2** | **45** | **-** | **-** | **-** | **-** |
| **3** | **5** | **-** | **-** | **-** | **-** |
| **3** | **5** | **-** | **-** | **1** | **5** |
| **2** | **45** | **2** | **9** | **-** | **-** |
| **1** | **245** | **3** | **11** | **-** | **-** |

Отже, вручну розрахувавши, бачимо, що алгоритм працює. Сума та кількість підраховані правильно.

**Висновок:**

Під час лабораторної роботи ми дослідили особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм. Математична модель, блок-схеми, псевдокод, код випробовування наведені. Оскільки перевірені вручну результати розрахунку правильні, то алгоритм правильно подає результат.