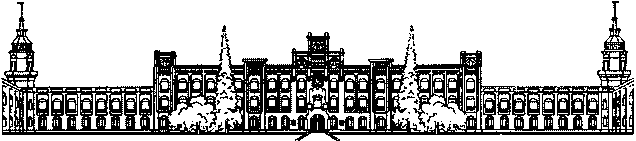
****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №1

**Теорія алгоритмів**

*«Вступ до алгоритмізації»*

Варіант 3

|  |  |  |
| --- | --- | --- |
| Виконали  студенти групи ІА-11: |  | Перевірив: |
|  |  |  |
| Момот А.Р. |  |  |
| Юрченко В.І |  |  |
| Старовойтов В.П.  Щур С. В. |  | ас. Цимбал С. І. |

Київ 2022

*Мета роботи:* навчитись використовувати рекурсію на простих прикладах.

*Хід роботи:*

*Завдання 1. Реалізувати задачу двома способами: рекурсивно та ітеративно.*

1. Постановка проблеми.

Обчислити суму елементів одновимірного масиву.

2. Побудова моделі.

Масив може складається з чисел: від’ємних, додатних, цілих або ж з плаваючою крапкою. Так як ми програмуємо мовою Python, яка є не типізованою. Ми сткикаємося з проблемою, що в функцію можуть передати масив з символами, тому нам потрібно робити перевірку на те що нам передали масив з числами.

3. Розроблення алгоритму.

Алгоритм обчислення суми елементів одновимірного масиву ітеративно

Вхідні дані:

Одновимірний масив з будь-яких чисел.

Вихідні дані:

Сума елементів одновимірного масиву.

1. Створюємо змінну sum, у яку ми запишемо суму елементів, та присвоємо їй значення 0.
2. Послідовно розглядаємо кожен елемент множини А.
3. Якщо елемент не останній, виконуємо операцію додавання до змінної sum.
4. Якщо елемент останній, виконуємо операцію додавання додавання до змінної sum i друкуємо її.

Алгоритм обчислення суми елементів одновимірного масиву рекурсивно

Вхідні дані:

Одновимірний масив з будь-яких чисел.

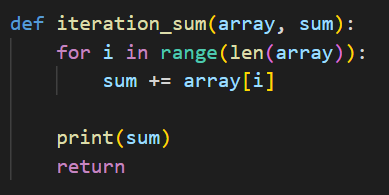
Вихідні дані:

Сума елементів одновимірного масиву.

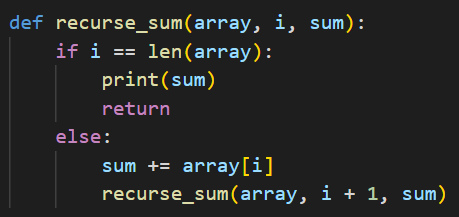
1. Створюємо змінну sum, у яку ми запишемо суму елементів, та присвоємо їй значення 0.
2. Рекурсія:
3. Якщо елемент не останній, додаємо його до змінної sum і переходимо до наступного елемента.
4. Якщо елемент останній, додаємо його до змінної sum i рекурсія припиняється.
5. Друкуємо суму.

4. Перевірка правильності алгоритму.

5. Реалізація алгоритму.



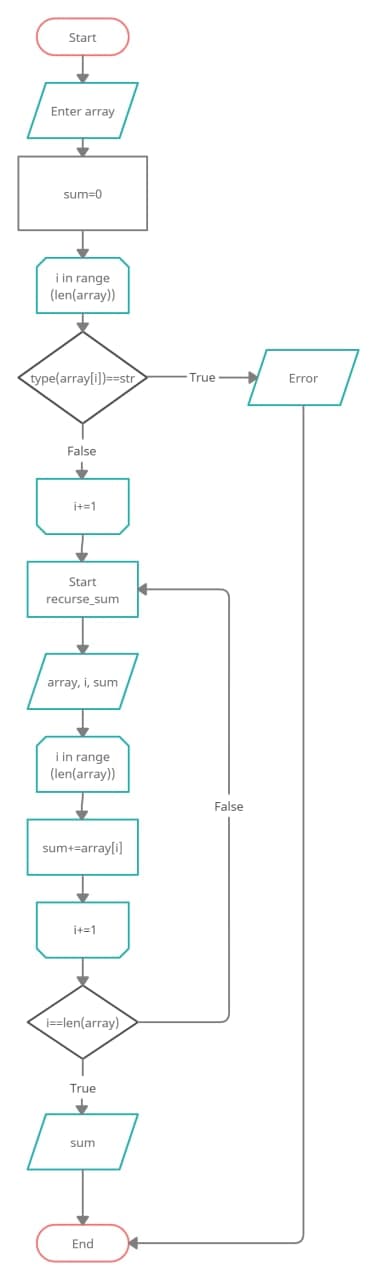
Ітеративна функція



Рекурсивна функція

6. Перевірка програми.

7. Складання документації (блок-схема алгоритму).

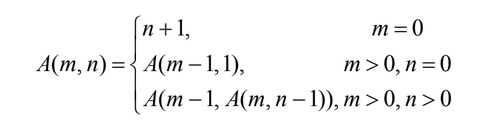


*Завдання 2. Написати рекурсивну функцію.*

1. Постановка проблеми.

Обчислити декілька значень функції Акермана для невід’ємних чисел m та

n:



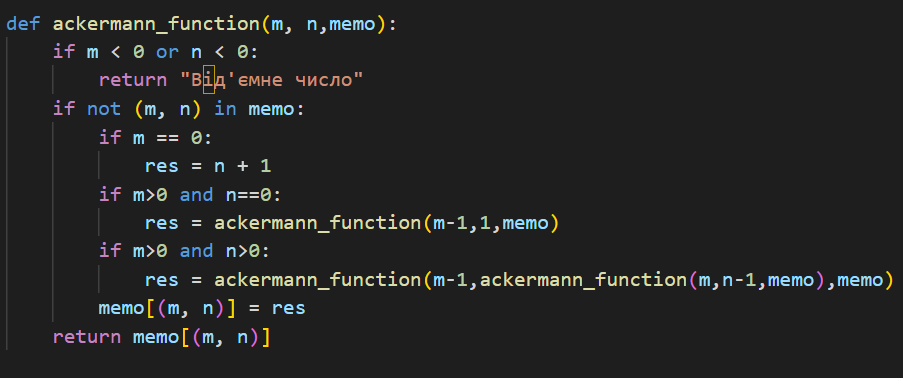
2. Побудова моделі.

Числа m та n повинні бути невід’ємними. Так як при m більше 0 та n більше 0 викликається рекурсія рекурсії це буде забирати дуже багато часу. Тому потрібно один із способів оптимізації, а саме меморізацію.

3. Розроблення алгоритму.

4. Перевірка правильності алгоритму.

5. Реалізація алгоритму.



6. Перевірка програми.

7. Складання документації (блок-схема алгоритму).

