**Практична робота 02.** Розробка блок-схеми алгоритму виконання процесу

**Мета:**

1) навчитися робити декомпозицію задачі;

2) навчитися надавати в графічній формі схему алгоритму виконання задачі.

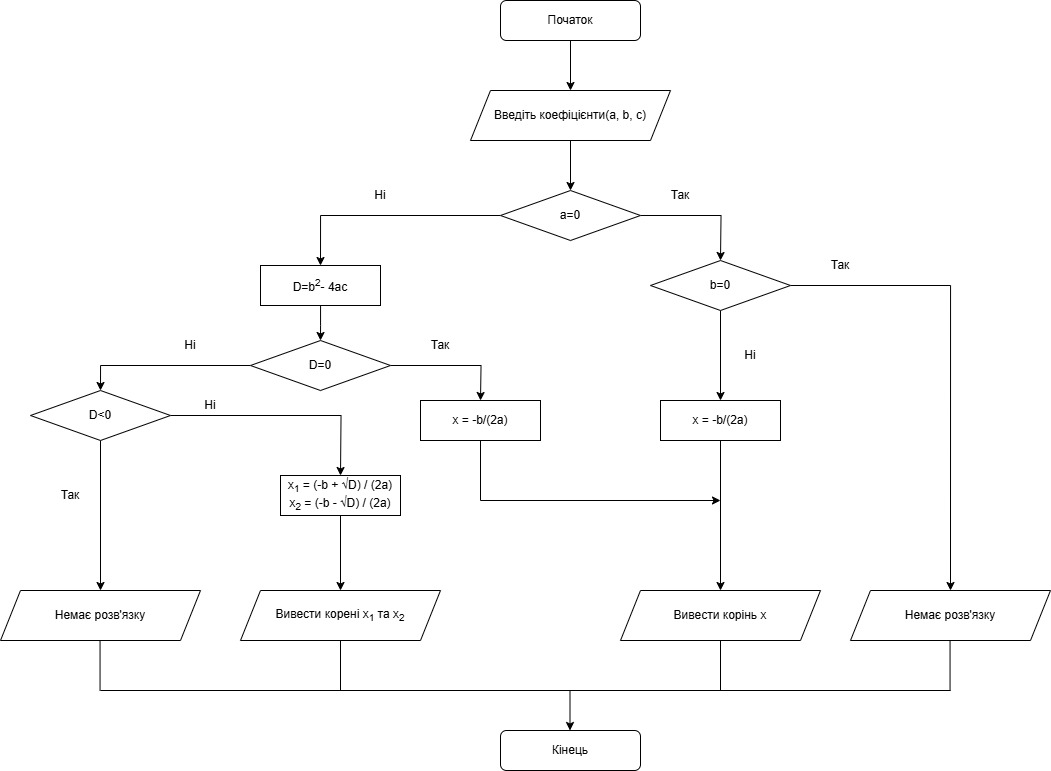
**Постановка задачі.**

Потрібно створити додаток/застосунок, який забезпечує введення, накопичення, коригування та оброблення інформації по викладачам і студентам гіпотетичного навчального закладу, а також пошуку інформації по викладачах та студентах. Введення може виконуватися в інтерактивному режимі або з файлу. Оброблення передбачає формування списків студентів по групах, викладачів по групах та предметах, звітності з успішності.

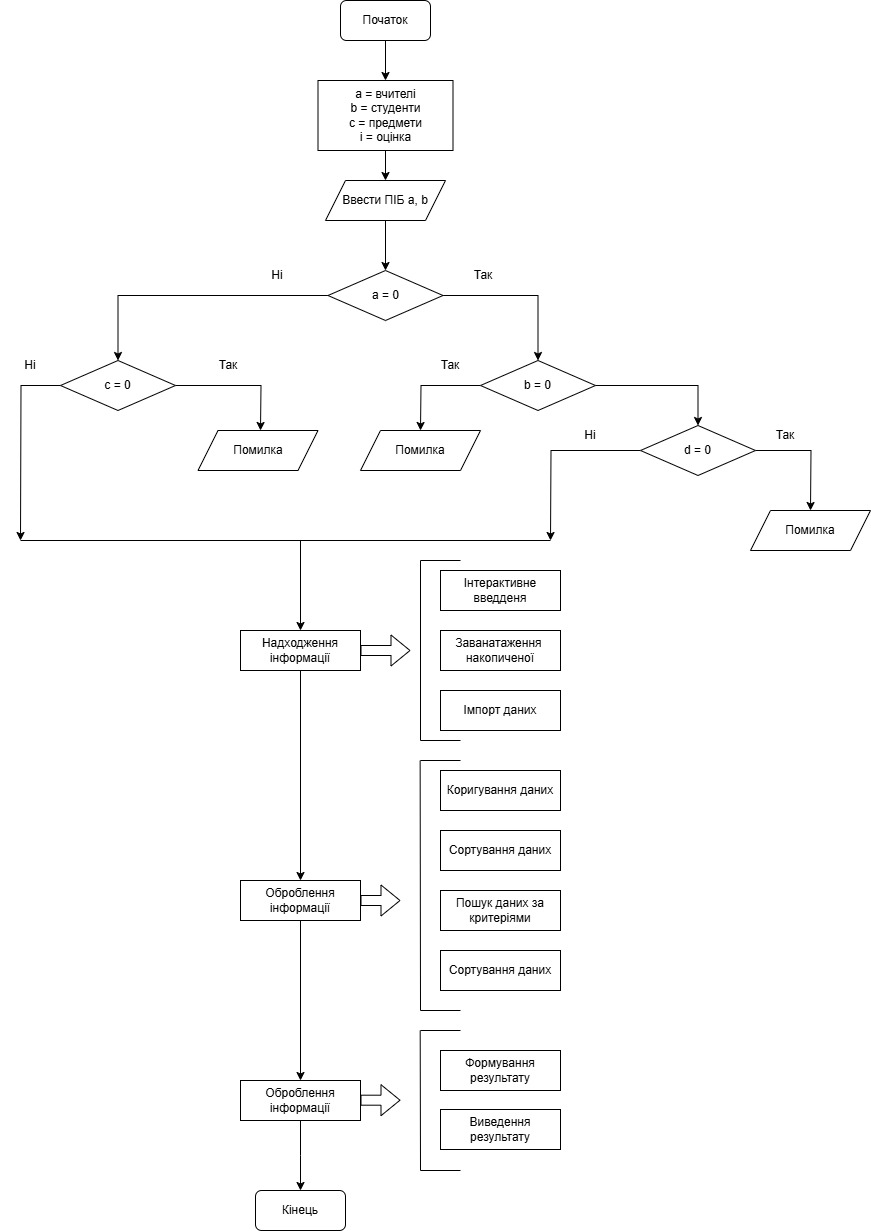
**Завдання практичної роботи**

1. Повторити матеріал лекцій №2 та 3, та ознайомитися з теоретичним матеріалом, наданим до цієї лабораторної роботи
2. Визначити перелік питань, які з вашої точки зору потребують уточнення для успішного розроблення алгоритму.
3. Провести функціональну декомпозицію задачі, накреслити схему.
4. Розробити блок-схему алгоритму виконання процесу як послідовності дій для отримання результату. Таким чином буде показана узагальнена схема технологічного процесу, який має реалізовувати додаток, тобто схему послідовності дій, які повинні виконуватися в ході вирішення задачі.
5. Розробити та накреслити блок-схему розрахунку коренів квадратного рівняння, де враховується багаторазове введення коефіцієнтів, аналіз дискримінанта, наявність двох, одного кореня та відсутність рішення, виведення результату.
6. Схеми можна виконувати в будь-якому доступному засобі, або накреслити на папері і зробити фотографію. Включити скріншот або фотографію (jpg-файл) в файл Word з практичною роботою

**Блок-схема розрахунку коренів квадратного рівняння**



**Блок-схема оброблення інформації по викладачам і студентам**



**Контрольні запитання.**

1. Алгоритм— це кінцева послідовність однозначних розпоряджень, виконання яких дозволяє за допомогою кінцевого числа кроків отримати рішення завдання, однозначно визначається вихідними даними. Його властивості: дискретність, визначеність, виконуваність, скінченність, масовість, результативність, формальність
2. Виконавець алгоритму — це об'єкт (людина, комп'ютер, робот тощо), який може виконувати алгоритм. Система команд — це набір команд, які розуміє виконавець
3. Правила побудови: однозначність команд, логічна послідовність дій, коректність для всіх допустимих вхідних даних, наявність початку та завершення
4. Механічні алгоритми, гнучкі алгоритми, лінійні алгоритми, розгалужені, циклічні, допоміжні
5. Словесно-формульний опис, графічний опис, алгоритмічною мовою або мовою програмування
6. Овал — початок або кінець, паралелограм — введення або виведення даних, прямокутник — обчислення або операції, ромб — перевірка умови
7. Програма перетворюється в машинні коди через кілька етапів. Спочатку її пишуть мовою програмування, потім компілюють у проміжний код, транслюють у машинний код (байт-код) і виконують процесором
8. Алгоритмізація - це процес побудови алгоритму розв'язання задачі.
9. Процеси та етапи алгоритмізації: формулювання задачі, визначення вхідних і вихідних даних, розробка алгоритму, запис алгоритму у формі коду, тестування та налагодження
10. Декомпозиція — це розбиття складної задачі на простіші частини для полегшення її реалізації
11. Інтеграція застосовується, коли потрібно об’єднати розрізнені частини системи або коду в єдине ціле
12. До прикладу алгоритму можна віднести приготування яєчні, де треба притримуватися чіткої послідовності кроків, або алгоритм дій під час домедичної допомоги. До алгоритмізації можна віднести приготування чаю