

Міністерство освіти і науки України
Національний університет “Львівська Політехніка”



Лабораторна робота №4

Виконав:

Студент групи АП-11

Білий Анатолій Іванович

Прийняв:

Чайковський І.Б.

Львів-2024

“Основи алгоритмізації”

Мета роботи: ознайомитися з структурою алгоритмів, навчитися створювати блок-схеми різних видів алгоритмів.

Теоретичні відомості

Розв'язання будь-якої задачі на ЕОМ відбувається в кілька етапів:

- формулювання постановки задачі;
- конструювання алгоритму розв'язання задачі;
- складання програми за розробленим алгоритмом;
- введення в ЕОМ програми і вихідних даних;
- налагодження і тестування програми;
- отримання розв'язку та аналіз результатів.

Алгоритм – це кінцева послідовність чітко визначених дій, які призводять до однозначного вирішення поставленого завдання.

Головна особливість будь - якого алгоритму – формальне виконання, що дозволяє виконувати задані дії– (команди) не тільки людині, але і різним технічним пристроям (виконавцям).

Процес складання алгоритму називається алгоритмізацією.

Розрізняють такі способи подання алгоритмів: текстовий, операторний і графічний. Найбільше поширення в наш час одержав графічний спосіб, при якому обчислювальний процес розчленовується на окремі операції, що відображаються у вигляді умовних графічних символів (блоків).

Теоретично доведено, що будь-який складний алгоритм може бути зведений до сукупності трьох елементарних:

- алгоритму слідування (лінійного);
- алгоритму розгалуження;
- алгоритму циклу.

Можна застосовувати довільну кількість рівнів вкладеностей, але логіка комп'ютерної програми повинна залишатися зрозумілою.

Загальний напрямок блок-схеми алгоритму прийнято формувати зверху-вниз.

Логіка послідовності дій зображується на блок-схемах лініями, причому для напрямків зверху-вниз і справа-наліво стрілки є необов'язковими.

Виконувані дії зображують прямокутниками, всередині яких пишуть пояснення дій (формули).

Для розгалужень і циклів використовують предикати (логічні вирази, або вирази порівняння, результатом виконання яких може бути тільки відповідь логічного змісту – «Так», «Ні», або «1», «0»).

Предикат зображують ромбом, до верхнього кута якого підходить лінія входу, а два інші кути використовують для відводу вихідних ліній альтернативної логіки. Всередині ромбу пишуть предикат, а лінії розгалуження підписують «Так», «Ні».

Деякі спеціальні дії зображують іншими фігурами, наприклад, ввід даних - паралелограмом, друк – імітацією листка паперу.

Найпростішими для розуміння і використання є лінійні структури.

Лінійним називають алгоритм (фрагмент алгоритму), в якому окремі команди виконуються послідовно одна за іншою, незалежно від значень вхідних даних і проміжних результатів.

Приклад лінійного алгоритму – алгоритм обчислення обсягу (V) прямокутного циліндра по радіусі (r) основи і висоті (h). Блок-схему показано на рис. 1.

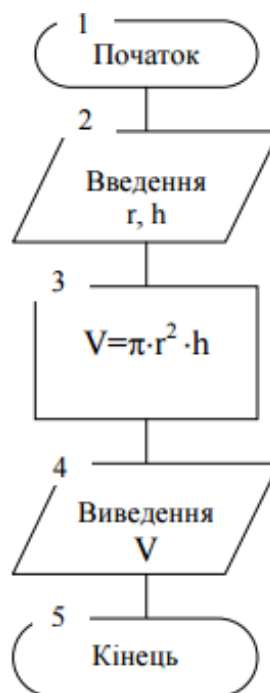


Рис. 1. Приклад лінійного алгоритму

Алгоритм лінійної структури реалізується в такий спосіб. Початок обробки даних – блок 1. Для проведення обчислень здійснюється введення в блоці 2 вихідних даних (значень r і h). У блоці 3 обчислюється обсяг циліндра V . Після обчислень здійснюється виведення результату (блок 4) і зупинка (блок 5).

Однак, розв’язання абсолютної більшості інженерних задач неможливо представити лише за допомогою лінійних алгоритмів.

Алгоритм, що розгалужується. Як правило, обчислювальний процес передбачає декілька можливих шляхів розв’язання задачі, реалізація яких залежить від виконання визначених умов.

Алгоритм, що розгалужується, (або просто розгалуження) застосовується в тих випадках, коли в залежності від умови необхідно виконати одну або іншу групу дій. На рис. 2 показано блок-схему алгоритму, що розгалужується. Окремий випадок розгалуження – обхід, коли по гілці «ні» ніяких дій виконувати не треба (блок-схема обходу – на рис.3).

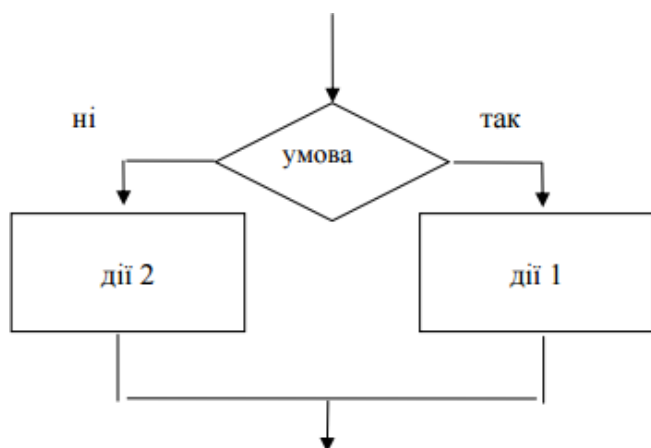


Рис. 2. Блок-схема розгалуження

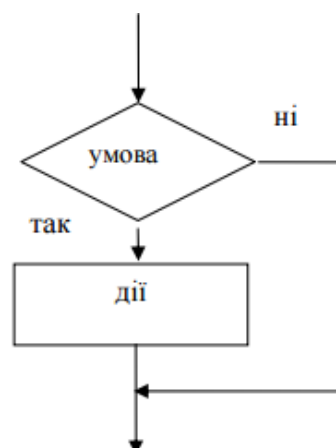


Рис. 3. Блок-схема обходу

Приклад. Обчислити значення f по одній із трьох формул – у залежності від значення x :

$$f = \begin{cases} a \cdot x^2 - 2 \cdot x + 7, & x < -1 \\ \frac{x \cdot \sin(x)}{(x+5)^2}, & -1 \leq x \leq 5 \\ \frac{\sqrt[3]{x} - 1}{\sqrt[3]{x} + 2}, & x > 5 \end{cases}$$

Блок-схема алгоритму даної задачі наведена на рис. 4.

Для обчислення значення f потрібно перевірити дві з трьох взаємовиключних умов (для $x < -1$ і $x > 5$). Після введення вхідних даних (блок 2) перевіряється перша умова $x < -1$ (блок 3). Якщо вона виконується, то значення f визначається по першій гілці формули (блок 4). У протилежному випадку перевіряється кожна з умов, що залишилися, (вони взаємовиключні).

У даному випадку в блоці 5 перевіряється умова $x > 5$. Якщо вона виконується, то значення f визначається по третій гілці формули (блок 6), у протилежному випадку - по другій гілці в блоці 7.

У блоці 8 здійснюється виведення результату.

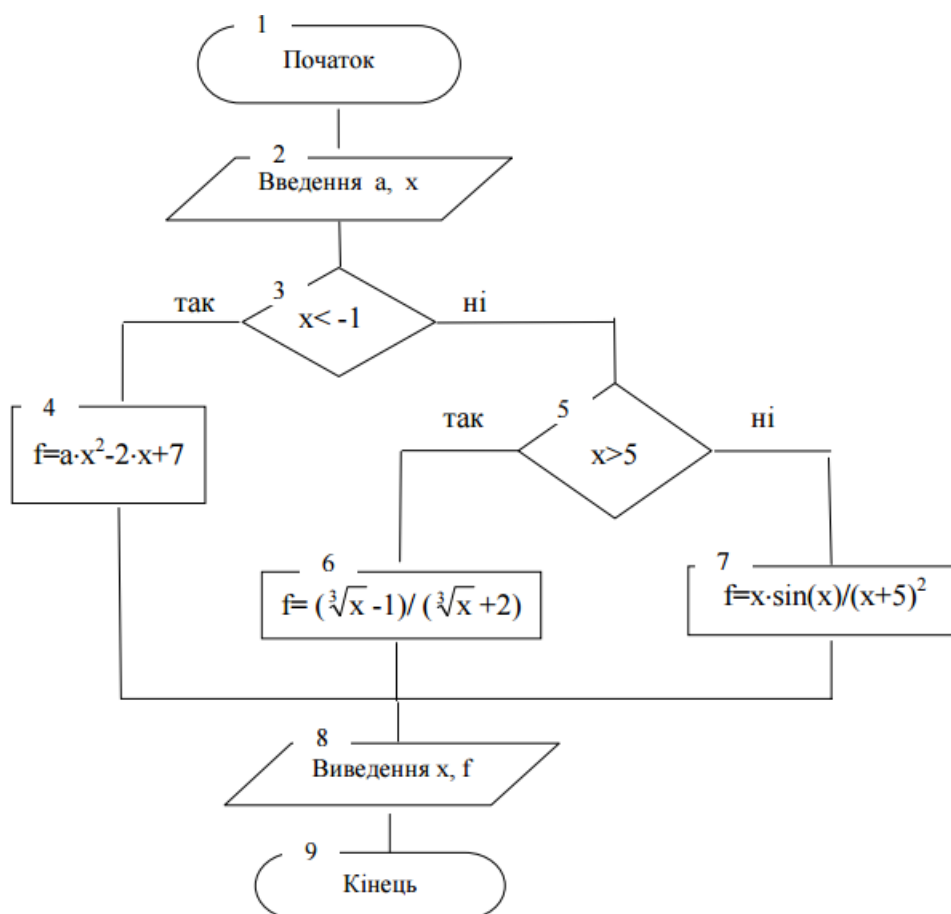


Рис. 4. Приклад розгалуженого алгоритму

Циклічний алгоритм. Алгоритм, окремі дії в якому багаторазово повторюються, називається циклічним (або просто циклом).

Багаторазово повторювані дії алгоритму називаються тілом циклу. Очевидно, що повторювати окремі обчислення доцільно при різних значеннях змінних. Одна з таких змінних називається керуючою змінною циклу.

Значення керуючої змінної визначає, чи буде цикл продовжуватися чи завершуватися.

Перед виконанням циклу необхідно присвоїти початкові значення керуючій змінній циклу i тим змінним, які будуть обчислюватися в циклі. Цей етап називається підготовкою циклу. Потім необхідно перевірити умову продовження циклу і задати правило зміни керуючої змінної для повторного виконання циклу. По числу повторень цикли поділяються на цикли з відомим числом повторень і цикли з невідомим числом повторень.

Цикли з відомим числом повторень. Це цикли, у яких керуюча змінна змінюється у відомих межах по відомому закону. Найпростіший випадок – коли керуюча змінна i змінюється від свого початкового значення i_n до кінцевого значення i_k із кроком Δi . Трійка величин $(i_n, i_k, \Delta i)$ називається параметрами циклу.

На рис. 5–7 представлені різні варіанти організації такого циклічного процесу. На рис. 5 показана організація циклу з постумовою; на рис. 6 – циклу з передумовою; на рис. 7 – циклу з блоком “модифікація”. Останні дві блок-схеми еквівалентні у тому сенсі, що реалізують той самий обчислювальний процес. Тому, щоб зрозуміти, як працює блок модифікації на рис. 7, досить звернутися до циклу з передумовою на рис. 6.

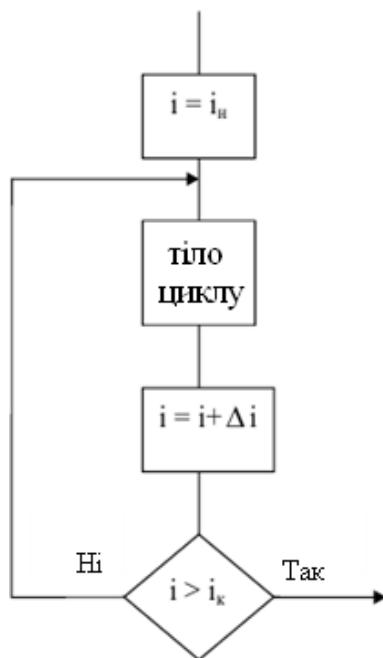


Рис. 5. Цикл із постумовою

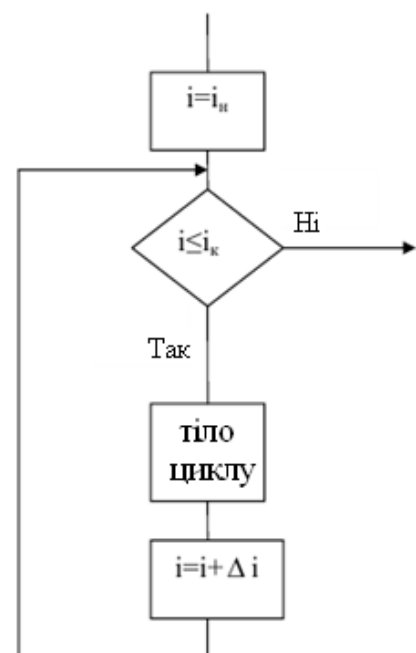


Рис. 6. Цикл із передумовою

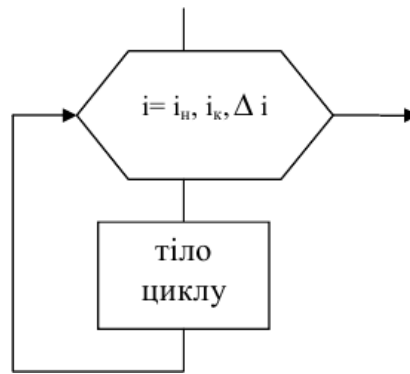


Рис. 7. Цикл із блоком “модифікація”

Якщо $\Delta i = 1$, то в блоці “модифікація” значення Δi не вказують. Кількість m повторень (ітерацій) циклу обчислюється по формулі: $m = [(i_k - i_n) / \Delta i + 1]$, де $[(i_k - i_n) / \Delta i + 1]$ - ціла частина величини $(i_k - i_n) / \Delta i$.

Цикли з невідомим числом повторень. У циклах з невідомим числом повторень число повторень циклу заздалегідь не визначено, а обчислювальний процес завершується, якщо виконуватиметься деяка умова. Щоб підрахувати кількість повторень циклу, необхідно організувати лічильник, який треба знулити до початку циклу. Цикли з невідомим числом повторень можуть бути двох типів – із передумовою (їх також називають циклами ПОКИ) і з постумовою (цикли ДО), які представлено на рис. 8 і 9 відповідно.

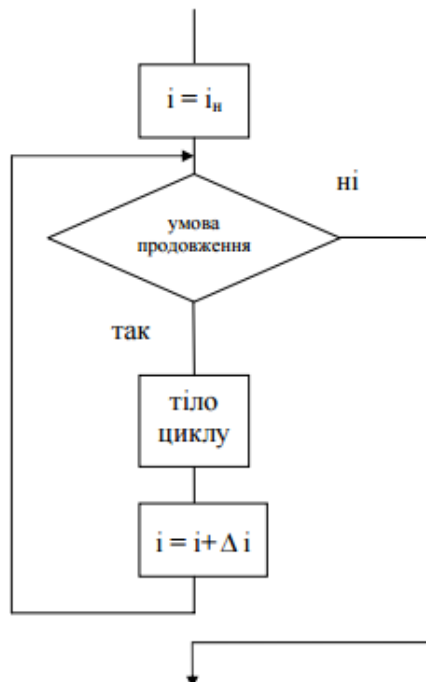


Рис.8. Цикл ПОКИ

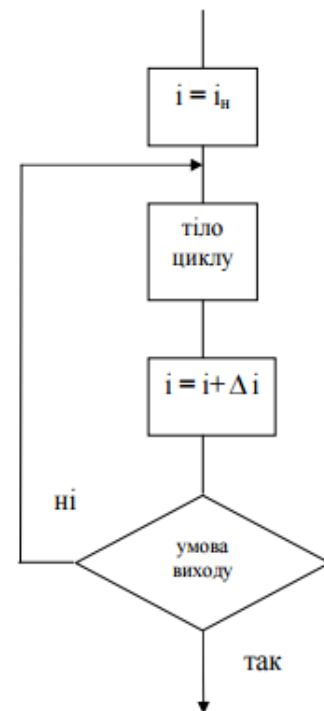


Рис. 9. Цикл ДО

Варто зауважити, що умови, які перевіряються в цих циклах, взаємно протилежні: у циклі ПОКИ перевіряється умова продовження циклу, а в циклі ДО – умова виходу з циклу.

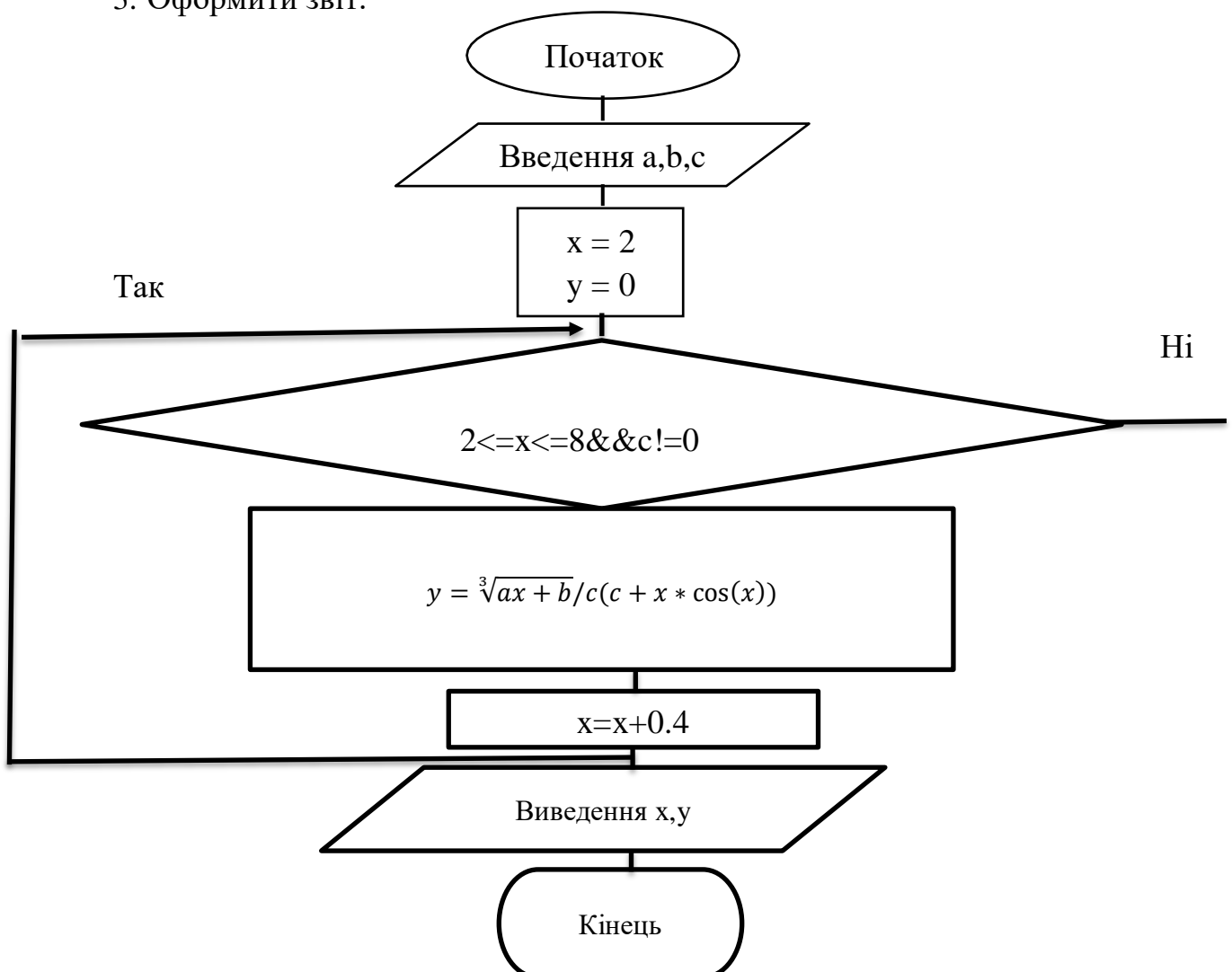
Особливість циклу ПОКИ: якщо при першій перевірці умова продовження порушується, те тіло циклу не буде виконано жодного разу.

Особливість циклу ДО: тіло циклу завжди виконується хоча б один раз.

В обчислювальному плані ці цикли еквівалентні, тобто в алгоритмі завжди можна замінити цикл ПОКИ циклом ДО і навпаки.

Хід роботи:

1. Ознайомитися з теоретичними відомостями.
2. Здійснити побудову у Microsoft Word блок-схеми розв'язку наступної функції, використовуючи циклічний алгоритм: $y = \sqrt[3]{ax+b} / c(c + x \cdot \cos(x))$ при $2 \leq x \leq 8$, $\Delta x = 0.4$. Значення a, b, c задані.
3. Оформити звіт.



Висновок: під час виконання цієї лабораторної роботи я ознайомився з структурою алгоритмів, навчився створювати блок-схеми різних видів алгоритмів.

Контрольні питання

1. Дайте визначення поняття «алгоритм».

Алгоритм — це впорядкована послідовність чітко визначених кроків або інструкцій, призначених для вирішення певної задачі або виконання конкретного завдання. Алгоритми можуть бути реалізовані у вигляді програмного коду, описані текстом або подані у вигляді блок-схем.

2. Які є способи подання алгоритмів ?

Алгоритми можна подати словесно або графічно. Перший спосіб - це словесний опис алгоритму. Другий спосіб - це подача алгоритму у вигляді таблиць, формул, схем, малюнків тощо. Третій спосіб - це запис алгоритмів за допомогою блок-схем.

3. Для чого використовують предикати?

Предикати використовуються для формулювання логічних висловлювань, які можуть бути істинними або хибними залежно від значень їх змінних

4. Назвіть види елементарних алгоритмів.

Лінійні (послідовні), розгалужені (умовні), циклічні, ітераційні, рекурсивні, комбіновані.

5. Назвіть типи циклів з невідомим числом повторень.

Є такі типи як while та for.