

Лабораторна робота № 1

ОСНОВИ АЛГОРИТМІЗАЦІЇ. НАЙПРОСТІШІ АГОРИТМИ.

Мета: Опанувати основи, поняття та принципи побудови алгоритмів. Навчитися самостійно складати найпростіші алгоритми розв'язку математичних рівнянь.

Теоретичні відомості

Алгоритмом називається чітко і однозначно визначена послідовність дій, необхідна для досягнення поставленої мети. Використання обчислювальних машин як виконувачів таких дій висуває ряд вимог до алгоритмів. Алгоритми мають такі властивості:

- зрозумілість;
- однозначність;
- дискретність;
- масовість;
- скінченність;
- результативність;
- правильність.

Під *зрозумілістю* алгоритмів мають на увазі те, що вони містять вказівки, які зрозумілі виконавцям.

Під *однозначністю* алгоритмів розуміють єдиний спосіб трактування правил виконання дій і порядку їх виконання.

Під *дискретністю* алгоритмів розуміють можливість розбиття всієї послідовності дій на окремі елементарні дії, виконання яких людиною, чи машиною не викликає сумнівів.

Під *масовістю* алгоритмів розуміють можливість їх виконання для рішення цілого класу конкретних задач, що відповідають загальній постановці задачі.

Скінченність алгоритму означає, що для досягнення поставленої мети використовується обмежена кількість кроків.

Результативність полягає у тому, що при будь-якому наборі вхідних даних алгоритм забезпечує або отримання шуканого результату, або повідомлення про те, що результат не можна отримати.

Існує чотири способи представлення (подання) алгоритмів:

- вербальний (словесний);
- алгебраїчний (за допомогою літерно-цифрових позначень виконуваних дій);
- графічний;
- з допомогою алгоритмічних мов програмування.

Словесна форма запису алгоритмів являє собою перелік дій, викладених звичайною мовою, який використовується в різних інструкціях, призначених для виконання їх людиною.

Алгебраїчна форма використовує математичні вирази та формули, найчастіше застосовується у теоретичних дослідженнях фундаментальних властивостей алгоритмів.

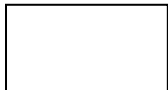

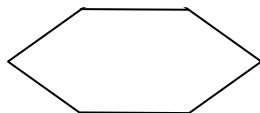
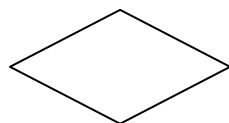

Графічна форма являє собою блок-схему алгоритму і відповідно до державних стандартів (ДСТУ) на оформлення документації, ця форма подання алгоритмів прийнята як основна для опису алгоритмів в офіційних документах.

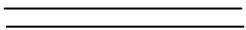
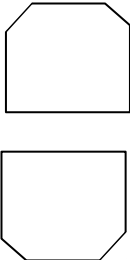




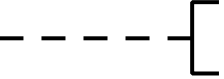

Алгоритм записаний з допомогою *алгоритмічної мови програмування* називається програмою. Алгоритм у такій формі може бути введений у комп'ютер і після відповідної обробки виконаний з метою отримання шуканого результату.


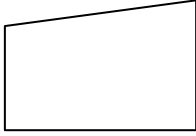

Схема – наочне графічне зображення алгоритму, коли окремі його дії (етапи) зображуються за допомогою різних геометричних фігур (блоків), а зв'язки між етапами указуються за допомогою стрілок, що сполучають ці фігури.

Схеми відображають кроки, які повинні виконуватися комп'ютером, і послідовність їх виконання. Умовні графічні позначення у схемах алгоритмів наведені у табл. 1.

Таблиця 1. Умовні графічні позначення у схемах алгоритмів

Назва символу та зміст позначення	Позначення
1. Процес Виконання операцій в результаті яких змінюються значення	
2. Зумовлений процес. Наперед визначений процес Виконання операцій у відповідності до окремо описаного алгоритму. Символ відображає зумовлений процес, що складається з однієї або декількох операцій чи кроків програми, які визначені в іншому місці (у підпрограмі, модулі). Тобто цей елемент використовується для позначення підпрограм.	
3. Модифікація. Підготовка Зміна значення змінної, яка керує виконанням циклічного алгоритму. Символ відображає модифікацію команди або групи команд з метою впливу на деяку подальшу функцію. Часто використовується для задання параметрів рахункового оператора циклу	
4. Розв'язання Перевірка умови та обрання напрямку виконання алгоритму. Символ відображає рішення або функцію типу перемикача, що має один вхід і ряд альтернативних виходів, один і лише один з яких може бути активізований після обчислення умов, визначених усередині цього символу. Відповідні результати обчислення можуть бути записані по сусідству з лініями, що відображають ці шляхи.	
5. Дані. Введення-виведення. Введення або виведення даних за допомогою стандартного пристрою введення-виведення. Символ відображає дані, носій даних не визначений. Ця фігура часто використовується і для запису різних алгоритмів, тому що для алгоритму джерело інформації не істотне і визначається тільки при реалізації програми.	

<p>6. Паралельні дії Виражає синхронізацію двох чи більше паралельних операцій</p>	
<p>7. Границя циклу Відображення початку та закінчення циклу</p>	
<p>8. Лінія Зображує потік даних або управління</p>	
<p>9. Пунктирна лінія Альтернативний зв'язок між символами. Використовують для обведення анотованої частини</p>	
<p>10. З'єднувач Позначення зв'язків між перерваними лініями потоку Символ відображає вихід з однієї частини схеми і вхід в іншу частину цієї схеми та використовується для обриву лінії і продовження її в іншому місці. Відповідні символи-з'єднувачі повинні містити одне і те ж унікальне позначення. Часто використовується при зображенні блок-схеми на декількох листах. Межа на одному листі позначається літерою A, а на іншому листі починається із з'єднувача, в якому також присутня літера A.</p>	
<p>11. Термінатор. Пуск-зупинка Початок та кінець алгоритму</p>	
<p>12. Коментар Символ використовують для додавання описових коментарів або записів пояснень у цілях пояснення або приміток. Пунктирні лінії в символі коментаря пов'язані з відповідним символом або можуть обводити групу символів. Текст коментарів або приміток повинен бути поміщений біля обмежувачої фігури.</p>	
<p>13. Пропуск Зображення пропуску символа або групи символів</p>	

<p>14. Документ</p> <p>Символ відображає дані, представлені на носіїв в легкій для читання формі (машинограма, документ для оптичного або магнітного зчитування, мікрофільм, рулон стрічки з підсумковими даними, бланки введення даних)</p>	
<p>15. Ручне введення даних</p> <p>Символ відображає дані, що обов'язково вводяться вручну під час обробки з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смужки зі штриховим кодом).</p>	
<p>16. Дисплей</p> <p>Символ відображає дані, представлені у формі, що легко читається людиною на спеціальному пристрої, який може бути екраном для візуального спостереження, індикатором виведення інформації тощо.</p>	

Інструкції з виконання будь-яких дій поміщаються всередину прямокутників (табл. 1, підпункт 1).

Раніше створені і окремо описані алгоритми і програми (а точніше підпрограми) зображуються у вигляді прямокутника з подвійними бічними лініями (табл. 1, підпункт 2). Усередині такого «подвійного» прямокутника вказуються ім'я алгоритму (підпрограми), параметри, при яких він повинен бути виконаний.

Шестикутник (табл. 1, підпункт 3) використовують для зміни параметра змінної, яка керує виконанням циклічного алгоритму, також зазначаються умови завершення циклу.

Блок вибору, що визначає шлях, по якому підуть ці дії (наприклад, обчислення) далі, залежно від результату аналізу даних, зображується у вигляді ромбу (табл. 1, підпункт 4). Сама умова записується усередині ромба. Якщо умова, що перевіряється, виконується, тобто має значення «істина», то наступним виконується етап по стрілці «так». Якщо умова не виконується («хибність»), то здійснюється перехід по стрілці «ні».

Введення початкових даних і виведення результатів зображуються паралелограмом (табл. 1, підпункт 5). Усередині нього пишеться слово «введення» або «друк» і перераховуються змінні, що підлягають введенню або виведенню.

Виконання декількох операцій одночасно, тобто паралельно, зображується за допомогою двох горизонтальних паралельних ліній (відрізків) (табл. 1, підпункт 6).

Умови для ініціалізації циклу, його приросту, завершення і т.д. розміщуються всередині символу (табл. 1, підпункт 7), на початку чи в кінці в залежності від розміщення операції, умови, яка перевіряється.

Для зображення напрямку потоку даних чи управління використовують лінію (табл. 1, підпункт 8). За необхідності чи для підвищення зрозумілості алгоритму можуть використовуватися стрілки, які вказують напрямок функціонування алгоритму.

Пунктирну лінію (табл. 1, підпункт 9) використовують для виділення анотованої частини програми та для зображення альтернативного зв'язку між символами алгоритму.

Стрілками зображуються можливі шляхи алгоритму, а малими кружечками (табл. 1, підпункт 10) – розриви цих шляхів (ліній) потоку на сторінці.

Початок і кінець алгоритму зображуються за допомогою овалів (табл. 1, підпункт 11). Усередині овалу записується “початок» або “кінець».

Коментарі використовуються в тих випадках, коли пояснення не поміщається усередині блока (табл. 1, підпункт 12).

Символ (три крапки) (табл. 1, підпункт 13) використовують в схемах для відображення пропуску символу або групи символів, для яких не визначені ні тип, ні кількість символів. Його використовують лише в символах лінії чи між ними. Найчастіше він використовується в схемах, що відображують загальний розв'язок з невідомим числом повторень.

Існують такі правила графічного запису алгоритмів:

- блоки алгоритмів з'єднуються лініями потоків інформації;
- лінії потоків не повинні перетинатися;
- будь-який алгоритм може мати лише один блок початку і один блок кінця;
- для полегшення аналізу і опису алгоритмів блоки можуть нумеруватися.

Номери проставляються у розриві контуру блока у верхній лівій частині.

Будь-який, навіть найскладніший, алгоритм може бути поданий у вигляді комбінацій кількох елементарних фрагментів, які називають базовими структурами. До них належать:

- послідовне проходження (рис. 1);
- розгалуження «якщо-то» (рис. 2);
- розгалуження «якщо-то-інакше» (рис. 3);
- обирання варіанта за ключем (рис. 4);
- цикл з параметром (рис. 5);
- цикл з передумовою (рис. 6); - цикл з післяумовою (рис. 7).
- використання коментаря (рис. 8);
- цикл з передумовою або післяумовою (рис. 9).

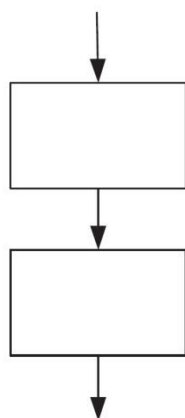


Рис. 1. Послідовне проходження

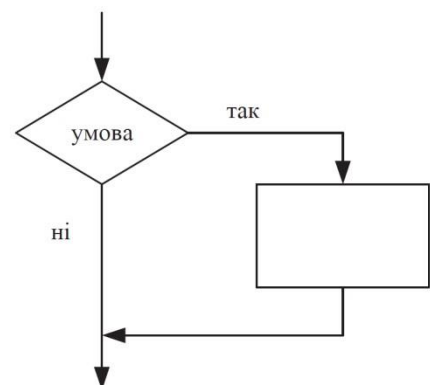


Рис. 2. Розгалуження «якщо-то»

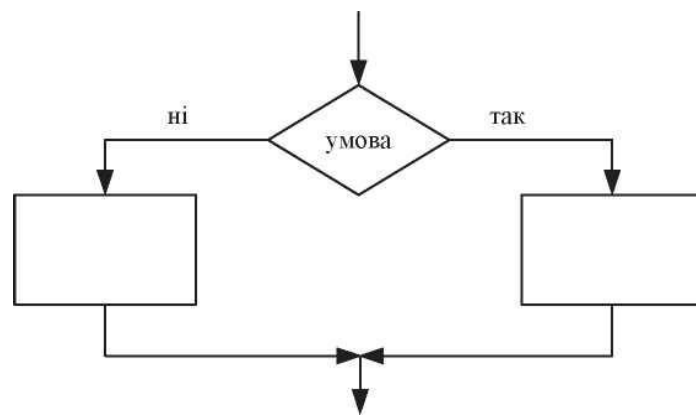


Рис. 3. Розгалуження «якщо-то-інакше»

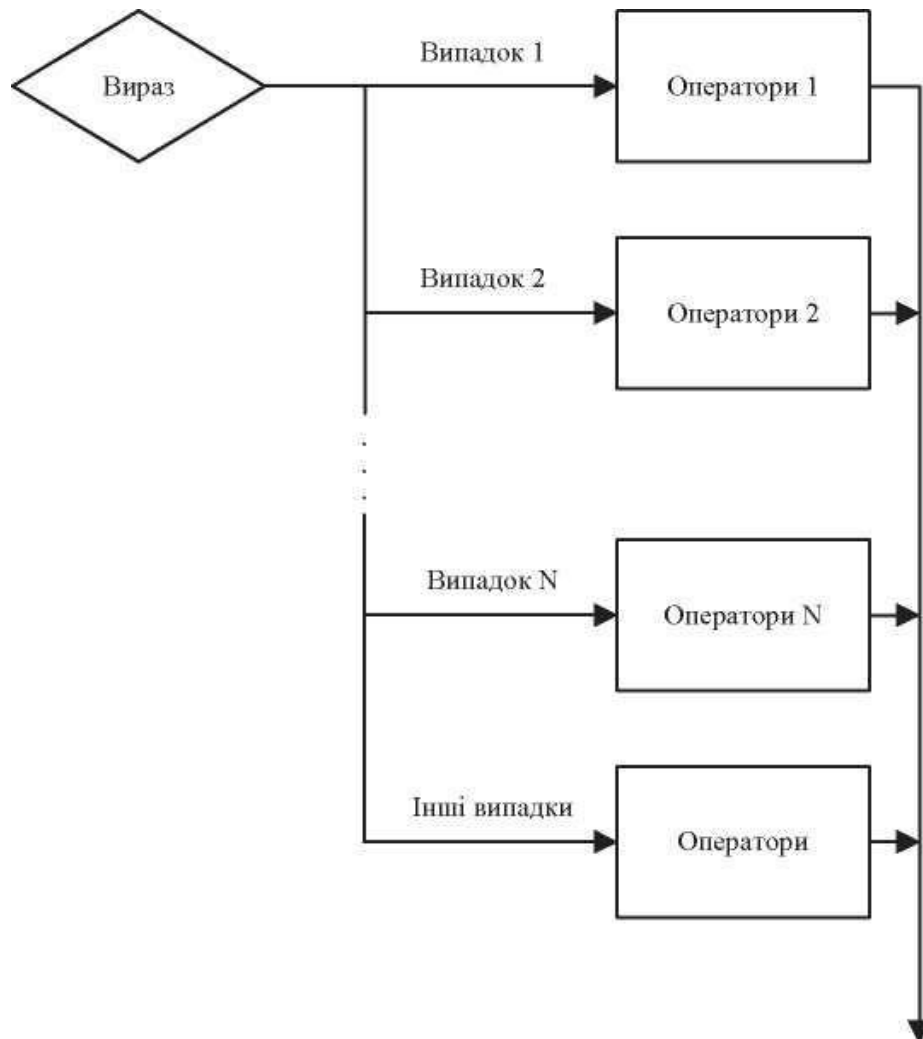


Рис. 4. Обирання варіанта за ключем

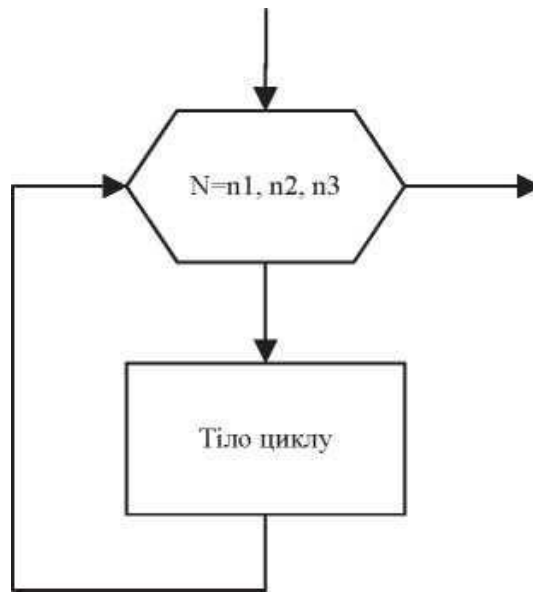


Рис. 5. Цикл з параметром

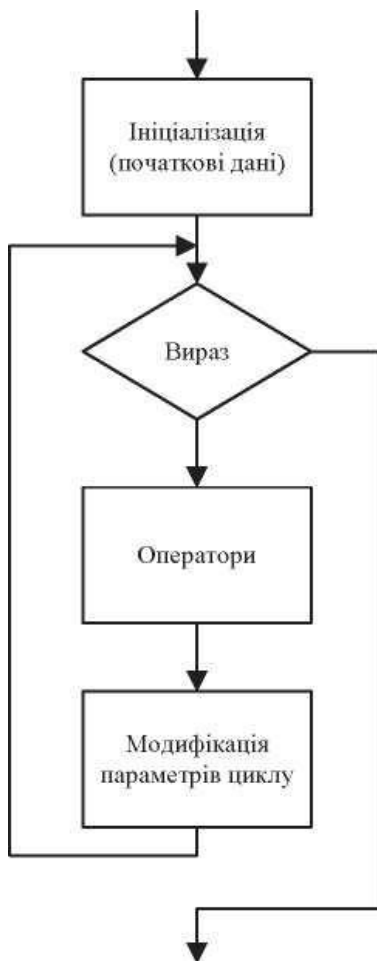


Рис. 6. Цикл з передумовою



Рис. 7. Цикл з післяумовою

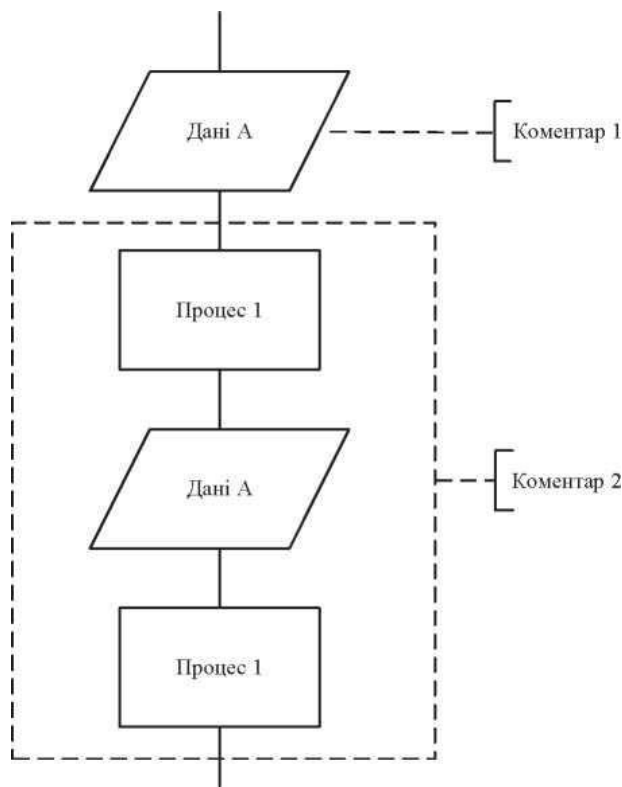


Рис. 8. Використання коментарів

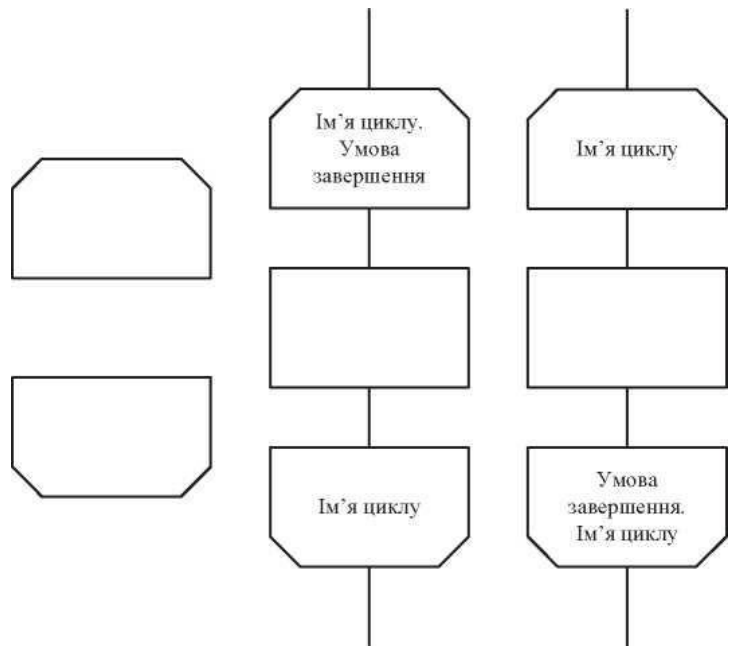


Рис. 9. Цикл з передумовою або післяумовою

Приклад. 1. Як приклад опишемо алгоритм природною мовою. Розглянемо алгоритм для знаходження коренів квадратного рівняння.

1. Почати.
2. Введення a, b, c .
3. D присвоїти $b^2 - 4ac$.
4. Якщо $D < 0$, то йти до п. 6.
5. Якщо $D > 0$, то йти до п. 8.
6. Дійсного коріння немає.
7. Перейти до п. 10.
8. Обчислити: $x_1 = (-b - \sqrt{D})/2a$, $x_2 = (-b + \sqrt{D})/2a$.
9. Вивести значення x_1 і x_2 .
10. Закінчити.

Кожному оператору відповідає стандартний блок з табл. 1 (ці умовні оператори замінюються конкретними командами мови програмування, на яку перекладається алгоритм).

1. Оператор 1 – початок роботи;
2. Оператор 2 – введення початкових даних;
3. Оператор 3 – початкове присвоювання;
4. Оператори 4, 5 – перевірка умови;
5. Оператори 6, 8 – дії;
6. Оператор 7 – лінія шляху виконання;
7. Оператор 9 – виведення результатів;
8. Оператор 10 – закінчення роботи.

В остаточному вигляді схема програми зображена на рис. 10. Альтернативний алгоритм розв'язання рівняння див. на рис. 10а.

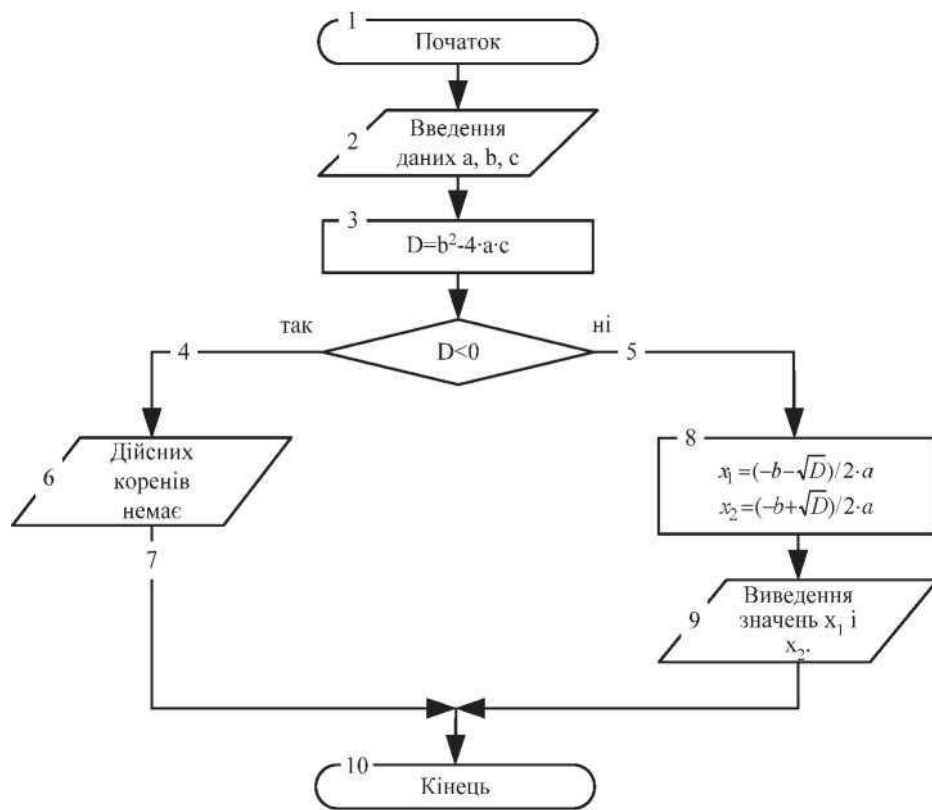


Рис. 10. Алгоритм програми знаходження коренів квадратного рівняння

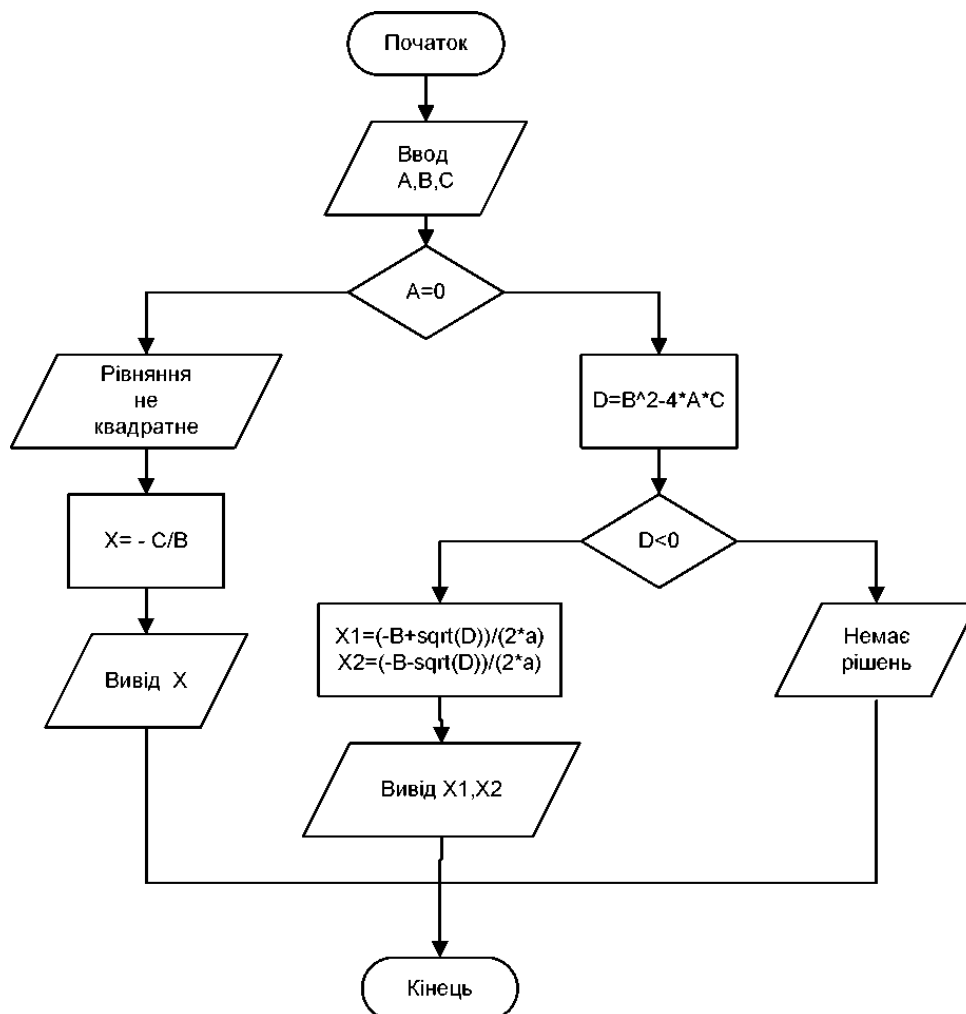


Рис. 10а. Покращений алгоритм програми знаходження коренів квадратного рівняння

Приклад. 2. Скласти схему програми, яка виводить на екран таблицю значень функції $f(x) = 2x^2 - 5x - 8$ в діапазоні від -4 до 4 . Крок зміни аргументу $0,5$.

Введемо змінні, які будуть використані у програмі.

Вхідні дані:

a – ліва границя відрізка, змінна дійсного типу;

b – права границя відрізка, змінна дійсного типу;

n – кількість вузлів, в яких обчислюється значення заданої змінної, змінна цілого типу.

Вихідні дані:

x – аргумент функції $f(x)$, змінна дійсного типу;

y – значення функції $f(x)$, змінна дійсного типу;

Проміжні дані:

dx – крок зміни аргументу, змінна дійсного типу;

i – параметр циклу, змінна цілого типу.

Опишемо алгоритм природною мовою.

1. Почнемо.
2. Оголосимо змінні, які будуть використані у програмі.
3. Вводимо з клавіатури значення змінних a , b та dx .
4. Обчислюємо кількість вузлів n , в яких будемо обчислювати значення аргументу.
5. Початковому значенню аргументу – змінній x присвоюємо значення лівої границі a .
6. Задаємо початкове значення лічильника циклу $i = 1$.
7. Задаємо цикл з умовою $i \leq n$.
8. Якщо вираз $i \leq n$ приймає значення “так» (true) переходимо до п. 10.
9. Якщо вираз $i \leq n$ приймає значення “ні» (false) переходимо до п. 14.
10. Обчислюємо значення функції $f(x) = 2x^2 - 5x - 8$.
11. Виводимо на екран значення змінних x та y .
12. Збільшуємо на крок dx поточне значення аргументу x .
13. Збільшуємо значення лічильника циклу i на 1.
14. Закінчимо.

Для графічного подання алгоритму використаємо стандартні блоки з табл. 1 та базові структури розглянуті на рис. 1–9. Згідно з розглянутим вище вербальним поданням алгоритму формуємо графічне подання у вигляді схеми програми, яка зображена на рис. 11.

В схемі програми, зображеної на рис. 11, використано декілька базових структур розглянутих на рис. 1–9. Крім послідовного проходження (див. рис. 1) використаний цикл з передумовою (див. рис. 6). Крім цього, для блока виведення на екран значень x та y , показаний приклад застосування коментаря (див. рис. 8).

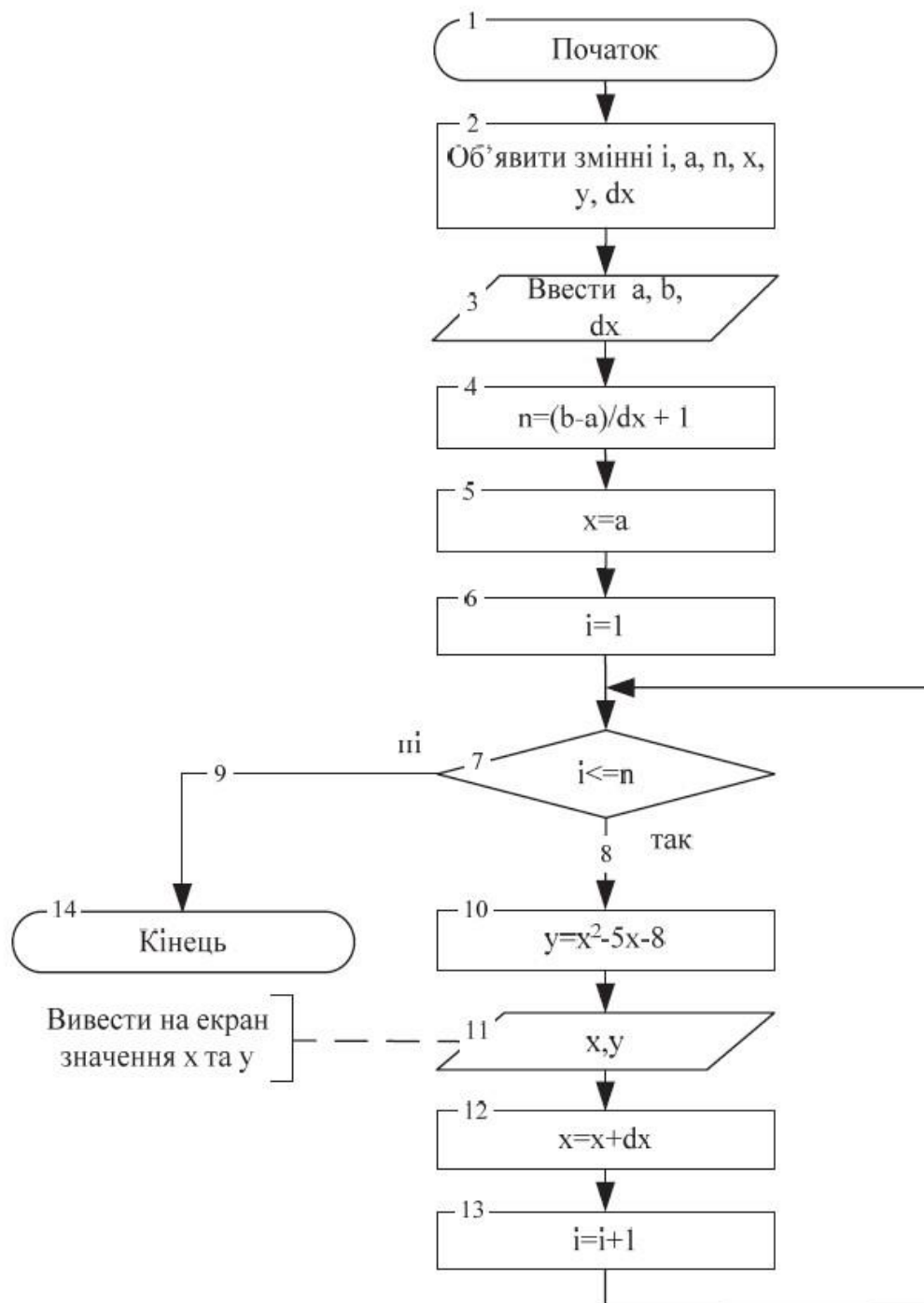


Рис. 11. Алгоритм програми табулювання функції

Приклад. 3. Скласти схему програми, яка обчислює середнє арифметичне послідовності дробових чисел. Після введення останнього числа програма повинна вивести на екран мінімальне та максимальне число послідовності. Кількість чисел послідовності задається під час роботи програми.

Введемо змінні, які будуть використані у програмі.

Вхідні дані:

a – поточне дійсне число послідовності, змінна дійсного типу;

n – кількість дійсних чисел, для яких обчислюється середнє арифметичне, змінна цілого типу.

Вихідні дані:

$sered$ – середнє значення послідовності, змінна дійсного типу;

max – максимальне значення послідовності, змінна дійсного типу;

min – мінімальне значення послідовності, змінна дійсного типу;

Проміжні дані:

sum – сума послідовності чисел;

i – параметр циклу, змінна цілого типу.

Опишемо алгоритм природною мовою.

1. Почнемо.
2. Оголосимо змінні, які будуть використані у програмі.
3. Присвоюємо нуль змінній sum .
4. Вводимо з клавіатури кількість чисел послідовності у змінну n , а також перше число послідовності у змінну a .
5. Ініціалізуємо змінні min , max та sum значенням змінної a .
6. Задаємо цикл з умовою $i < n$, початковим значенням $i = 1$, та збільшенням лічильника циклу на 1.
7. Якщо вираз $i < n$ приймає значення *true* (так) переходимо до п. 9.
8. Якщо вираз $i < n$ приймає значення *false* (ні) переходимо до п. 19.
9. Вводимо з клавіатури поточне число послідовності у змінну a .
10. Додаємо до значення змінної sum значення змінної a .
11. Вводимо умову $a < min$.
12. Якщо вираз $a < min$ приймає значення *true* переходимо до п. 14.
13. Якщо вираз $a < min$ приймає значення *false* переходимо до п. 15.
14. Присвоюємо змінній min значення змінної a .
15. Вводимо умову $a > max$.
16. Якщо вираз $a > max$ приймає значення *true* переходимо до п. 18.
17. Якщо вираз $a > max$ приймає значення *false* переходимо до.
18. Присвоюємо змінній max значення змінної a .
19. Обчислюємо значення середнього арифметичного $sered$.
20. Виводимо на екран значення $sered$.
21. Закінчимо.

Для графічного подання алгоритму використаємо стандартні блоки з табл. 1 та базові структури, розглянуті на рис. 1–9. Згідно з розглянутим вище поданням вербального алгоритму формуємо графічне подання у вигляді схеми програми, яка зображена на рис. 12.

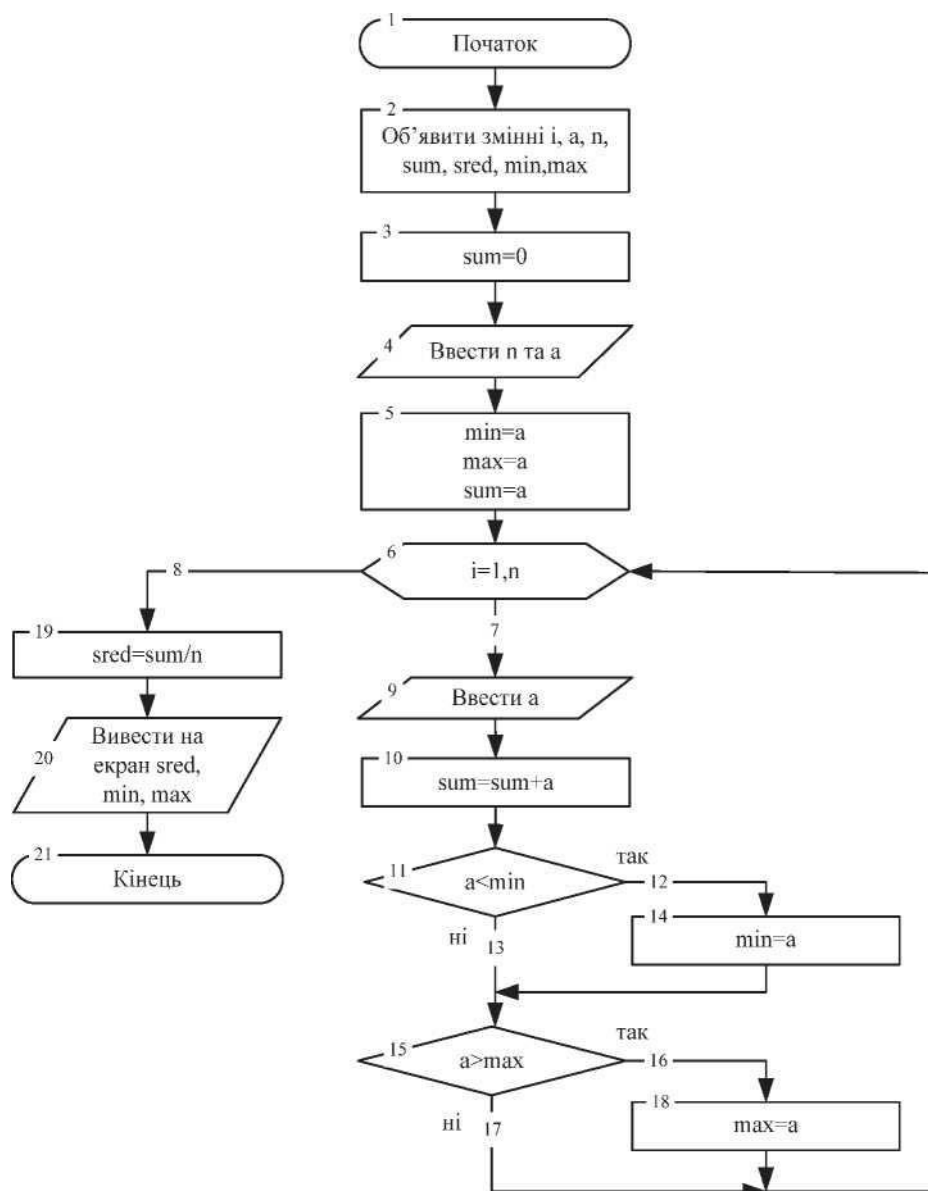


Рис. 12. Алгоритм програми, яка обчислює середнє арифметичне

В схемі програми зображеної на рис. 12 використано декілька базових структур розглянутих на рис. 1–9. Крім послідовного проходження (див. рис. 1) використаний цикл з параметром (див. рис. 5). Крім цього, для знаходження *min* та *max* використано розгалуження «якщо-то» (див. рис. 2).

ЗАВДАННЯ

1. Опрацювати і засвоїти матеріал наведений в теоретичних відомостях.
2. Записати у математичній формі площу трикутника, заданого довжинами його сторін.
3. Описати природною мовою послідовність дій для успішного обчислення площі трикутника, заданого довжинами його сторін.
4. Побудувати графічно блок-схему алгоритму програми обчислення площі трикутника, заданого довжинами його сторін.
5. У звіті подати результати виконання пунктів 2-4.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке алгоритм?
2. Наведіть властивості алгоритмів.
3. Що розуміють під дискретністю (зрозумілістю, ...) алгоритмів?
4. Які існують способи представлення алгоритмів?
5. Який спосіб подання алгоритмів прийнятий за основний?
6. Що таке схема алгоритму?
7. Які графічні символи використовуються у схемах алгоритмів?
8. Як позначають розриви в схемах алгоритмів?
9. Яким чином у схемах алгоритмів зображується символ «модифікація»?
10. Яким чином у схемах алгоритмів зображується символ «введення– виведення»?
11. Яким чином у схемах алгоритмів зображується символ «процес»?
12. Яким чином у схемах алгоритмів зображується символ «розв'язання»?
13. Яким чином у схемах алгоритмів зображується символ «коментар»?
14. Яким чином у схемах алгоритмів зображується символ «наперед визначений процес»?
15. Яким чином у схемах алгоритмів зображується символ «паралельні дії»?
16. Як на схемах зображають напрям потоку даних чи управління алгоритмом?
17. Які існують правила графічного запису алгоритмів?
18. Що таке базова структура?
19. Як зображується розгалуження «якщо-то « у схемах алгоритмів?
20. Як зображується розгалуження «якщо-то-інакше» у схемах алгоритмів?
21. Як зображується вибір за ключем у схемах алгоритмів?
22. Як зображується цикл з параметром у схемах алгоритмів?
23. Як зображується цикл з передумовою у схемах алгоритмів?
24. Як зображується цикл з післяумовою у схемах алгоритмів?
25. Наведіть приклад використання коментарю у схемах алгоритмів.
26. Зобразіть схему програми знаходження коренів квадратного рівняння.
27. Зобразіть схему програми обчислення середнього арифметичного масиву чисел.
28. Зобразіть схему програми табулювання функції.