**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

Кафедра спеціалізованих комп'ютерних систем

С. О. Бантюкова

АЛГОРИТМИ ТА ОСНОВИ ПРОГРАМУВАННЯ

Конспект лекцій

Частина 1

Харків – 2018

Бантюкова С. О. Алгоритми та основи програмування : Конспект лекцій. – Харків : УкрДУЗТ, 2018. – Ч.1. – 35 с.

Конспект лекцій розроблено відповідно до робочої програми дисципліни «алгоритми та основи програмування» для студентів факультету ІКСТ. До нього включено розділи, що необхідні студентам для опанування теоретичних та практичних основ застосування теорії алгоритмів для розв'язання широкого кола задач, використовуючи засоби обчислювальної техніки.

Рекомендується для студентів факультету ІКСТ. Іл. 27, табл. 1, бібліогр. : 4 назв.

Конспект лекцій розглянуто та рекомендовано до друку на засіданні кафедри **с**пеціалізованих комп’ютерних систем 12 лютого 2018 р., протокол № 10.

Рецензент

проф. А. О. Каргін

С. О. Бантюкова

АЛГОРИТМИ
ТА ОСНОВИ ПРОГРАМУВАННЯ

Конспект лекцій

Частина 1

Відповідальний за випуск Бантюкова С. О.

Редактор Решетилова В. В.

Підписано до друку 03.04.18 р.

Формат паперу 60х84 1/16. Папір писальний.

Умовн.-друк.арк. 2,0. Тираж 25. Замовлення №

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.

ЗМІСТ

1 ПОНЯТТЯ АЛГОРИТМУ. ТИПИ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ. ЛІНІЙНИЙ ОБЧИСЛЮВАЛЬНИЙ ПРОЦЕС

Розв’язання задачі на ЕОМ включає такі етапи:

- постановка задачі і вибір чисельного методу її розв’язання;
- розробка алгоритму;
- складання програми алгоритмічною мовою;
- реалізація програми на ЕОМ, включаючи дії з введення, відлагодження і виконання програми на ЕОМ;
- аналіз отриманих результатів.

ЕОМ є усього лише автоматом, що швидко і точно виконує приписи, складені людиною. Розробка таких приписів, тобто проектування ходу розв’язання задачі – невід'ємна частина діяльності, що пов'язана з використанням ЕОМ. На початковому етапі ці приписи подаються у вигляді алгоритму.

Алгоритм – це послідовність виконання кінцевого числа дій, необхідних для розв’язання задач заданого типу.

Властивості алгоритму:

- детермінованість (визначеність) – однозначність результату процесу при заданих вхідних даних;
- дискретність – поділ алгоритму на окремі елементарні дії, можливість виконання яких людиною або машиною не викликає сумнівів;
- масовість – можливість вибору вхідних даних з деякої множини даних;
- результативність – одержання результату або повідомлення про неможливість одержання результату при заданих вхідних даних.

За алгоритмом складається комп’ютерна програма розв’язання задачі.

Комп’ютерна програма – це опис алгоритму розв’язання задачі алгоритмічною мовою, реалізуючи яку ЕОМ перетворює вхідні дані на результат розв’язання задачі.

Засоби опису алгоритмів:

- словесний опис, при якому послідовність дій описується словесно;

- операторна схема, при якій використовується рядковий символьний запис з індексами для вказування заданої послідовності виконання дій;
- схема алгоритму, при якій обчислювальний процес зображується графічно за допомогою визначених блоків (таблиця 1).

Елементами схем алгоритму є геометричні фігури (блоки), кожна з яких визначає певну дію, та лінії потоку, що вказують послідовність виконання дій. Вид дії та дані, над якими виконуються дії, записуються в середині блоків алгоритму традиційним способом (формули, відношення і т. ін.). Схеми алгоритмів є універсальним способом запису алгоритмів, бо їхній вигляд не залежить від того, якою алгоритмічною мовою в подальшому вони будуть реалізовані. Ще одна перевага полягає у високій наочності схем. При побудові алгоритмів вибирається різноманітна глибина деталізації окремих операцій. На схемах алгоритмів стрілками позначаються тільки лінії потоку, що мають злам і напрямки, які вказують знизу вгору, праворуч – ліворуч. Розмір *a* фігури алгоритму вибирається з ряду 10, 15, 20 мм (таблиця 1). Ці розміри можна збільшити на число, кратне 5. Розмір *b=1.5*a.*

Для зручності читання схем алгоритмів блокам процесу присвоюють порядкові номери. Номери блоків процесів проставляються так, щоб їх можна було читати згори вниз та зліва–направо, незалежно від напрямку потоку.

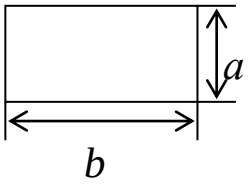
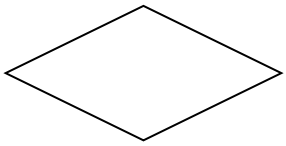
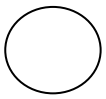
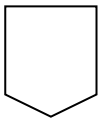
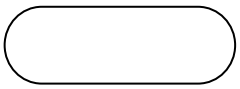
При зображенні схем алгоритмів слід керуватися єдиною системою програмної документації, до якої входять держстандарти зі схем алгоритмів та програм.

При складанні схем алгоритмів обчислювальних процесів виділяють такі типи обчислювальних процесів (типові структури алгоритмів):

- лінійний;
- розгалужений;
- циклічний.

Лінійний обчислювальний процес характеризується тим, що кроки (блоки), на які він розбивається, виконуються послідовно в тому порядку, в якому вони подані. У практиці програмування лінійні алгоритми в чистому вигляді практично не зустрічаються.

Таблиця 1 – Найменування, графічне зображення та функції блоків алгоритму

Найменування блоків	Графічне зображення	Функції
Процесс		Виконання операції або групи операцій
Виконання		Вибір напрямку виконання алгоритму в залежності від заданих умов
Модифікатор		Початок циклу
Введення, виведення		Введення даних, виведення результатів
Звернення до підпрограми		Використання раніше створених і окремо описаних алгоритмів
З’єднувач (сторінковий)		Розрив лінії потоку
З’єднувач (міжсторінковий)		Розрив лінії потоку з переходом на іншу сторінку
Початок, кінець алгоритму		Початок, кінець алгоритму, вхід, вихід в підпрограмах

Об'єктами опису в схемах алгоритмів є константи, змінні величини, вирази (арифметичні і логічні).

Константа – це величина, що має постійне значення, яке не змінюється при реалізації алгоритму. Наприклад, числа 7, –1.34, 0.00556 можуть бути константами.

Змінна – це величина, значення якої може змінюватися в процесі реалізації алгоритму. Змінна має своє власне ім'я – **ідентифікатор.** Змінна величина перед початком

обчислювальних операцій повинна бути визначена, тобто їй повинно бути задане конкретне значення операціями введення або присвоювання.

Приклад 1. Скласти схему алгоритму обчислення значення функції $Y = 3 + \frac{\ln(x^2+1)}{2x+a-5}$, якщо $x = \frac{a+b}{b^2}$, a = 3. 45, b – будь-яке число, що більше 1.

При складанні схеми алгоритму (рисунок 1) скористуємось способом обчислення частинами. Для цього позначимо чисельник символом H, знаменник – Z. Блоки 2, 3 – визначають змінні, блоки 4, 5, 6, 7 – розрахунок, блок 8 – виведення результатів. Всі блоки алгоритму розміщені послідовно.

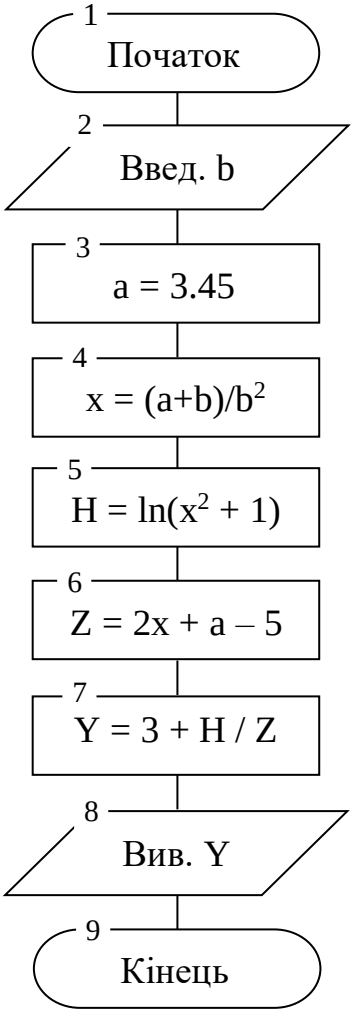


Рисунок 1 – Алгоритм обчислення значення функції Y

Контрольні запитання

1 Поняття алгоритму.
2 Властивості алгоритму.
3 Поняття комп'ютерної програми.
4 Засоби опису алгоритмів.
5 Блоки, які використовуються в схемах алгоритмів.
6 Типи обчислювальних процесів.
7 Поняття лінійного обчислювального процесу.
8 Поняття константи, змінної.

2 РОЗГАЛУЖЕНИЙ ОБЧИСЛЮВАЛЬНИЙ ПРОЦЕС

Розгалуженим називають алгоритм, в якому вибір дії залежить від виконання визначених умов, значень вхідних даних або проміжних результатів.

Алгоритми розгалужених обчислювальних процесів являють собою структури, що мають декілька варіантів обчислень. Кожний варіант подається окремою обчислювальною гілкою, вибір гілки здійснюється управляючою частиною алгоритму.

Управляюча частина алгоритму – блоки «рішення» (інша їх назва – логічні або умовні блоки) з'єднані так, щоб для визначеного набору вхідних даних або проміжних результатів гарантувалося виконання дій за єдиною гілкою. Вибір варіанта задається логічним відношенням або логічним виразом.

Логічне відношення – послідовний запис констант, змінних, арифметичних виразів, об'єднаних операціями відношення $>, <, \geq, \leq, =, \neq$.

Логічний вираз – послідовний запис логічних відношень, об'єднаних знаками логічних операцій:

- логічне множення або операція «І» позначається знаком $\wedge$;
- логічне додавання або операція «АБО» позначається знаком $\vee$;
- логічне заперечення або операція «НІ» позначається знаком **!** .

Загальний вигляд схеми розгалуженого обчислювального процесу наведено на рисунку 2. В залежності від результату перевірки умови виконується Дія 1 або Дія 2. Кожна з дій може являти собою складову структуру з будь-яких блоків.

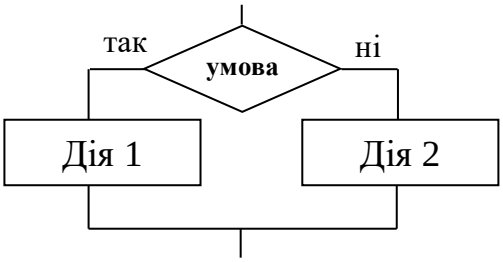


Рисунок 2 – Загальний вигляд схеми розгалуженого обчислювального процесу

Окремим випадком розгалуження є структура «обхід», в якій одна гілка не містить ніяких дій (рисунок 3).

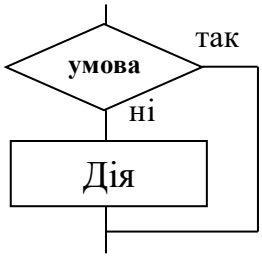


Рисунок 3 – Структура «обхід»

Узагальненням розгалуження є структура «множинний вибір», в якій одна з дій обирається в залежності від значення управляючої змінної (рисунок 4).

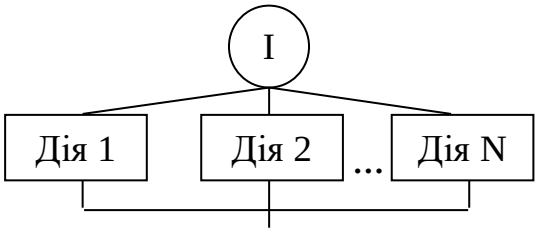


Рисунок 4 – Структура «множинний вибір»

Приклад 2. Скласти схему алгоритму обчислення значення функції Y:

$$Y = \begin{cases} 2x + 3, \text{ якщо } x < 0; \\ 4x - 7, \text{ якщо } x \geq 0. \end{cases}$$

Схему алгоритму наведено на рисунку 5.

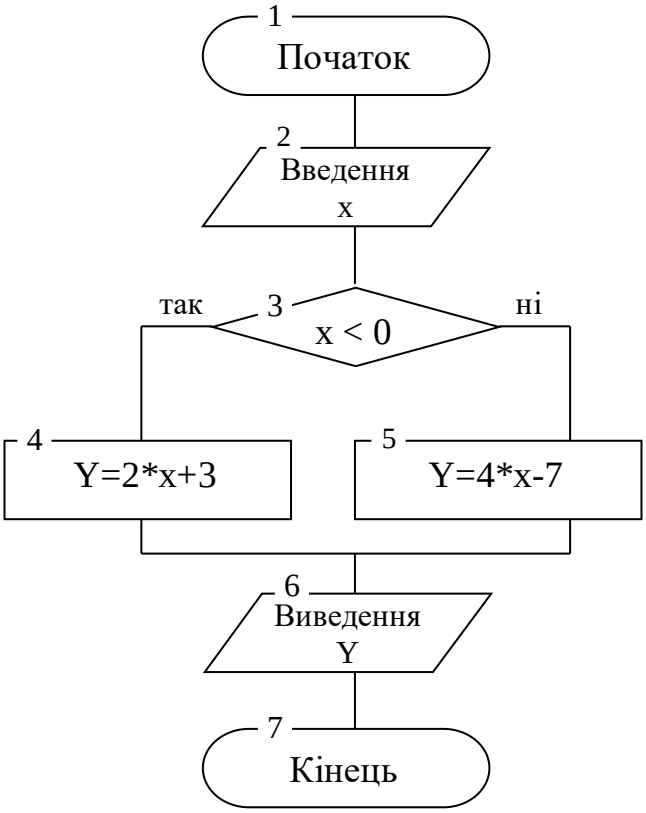


Рисунок 5 – Алгоритм обчислення значення функції

У прикладі вказано дві умови ($X < 0 \wedge X \geq 0$), але у алгоритмі достатньо записати одну з них: $X < 0$. Цей блок має два виходи: вихід на блок 4, якщо $X < 0$, та вихід на блок 5, якщо $X \geq 0$.

Приклад 3. Скласти схему алгоритму обчислення функції Z:

$$Z = \begin{cases} 2x + 3y, \text{якщо } 0 < X < 2 \ \text{ I } \ Y > 3; \\ \dfrac{X}{3} + Y, \text{якщо } X < 0 \ \text{ АБО } \ X > 6; \\ Y - X - 1, \text{якщо } X = 5; \\ X \cdot Y, \text{в інших випадках.} \end{cases}$$

Схему алгоритму наведено на рисунку 6.

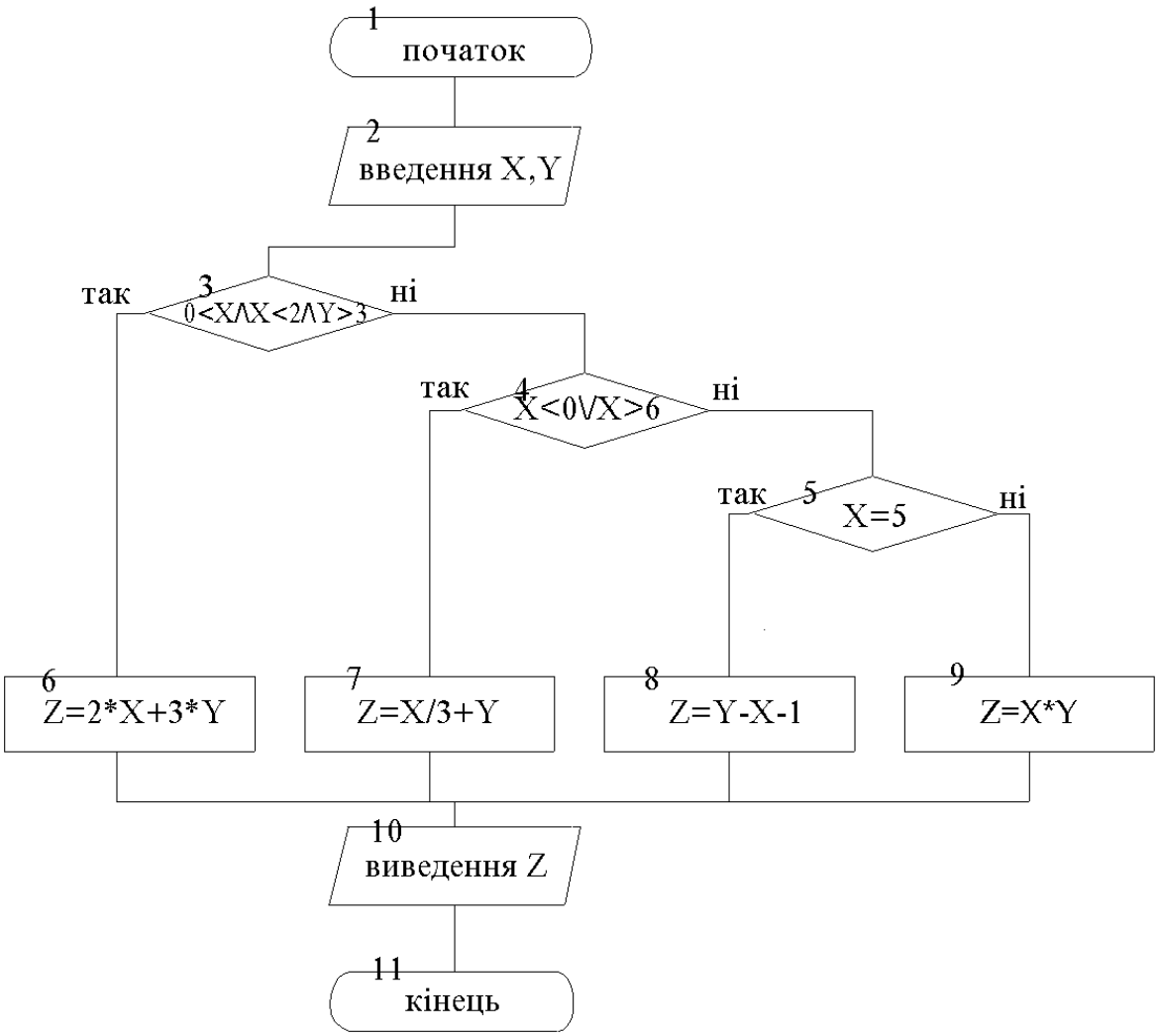


Рисунок 6 – Алгоритм обчислення функції Z

Управляюча частина схеми, яка забезпечує вибір одного з багатьох шляхів обчислення функції Y, подана трьома блоками «рішення» (блоки 3, 4, 5).

Логічний вираз можна аналізувати в межах як одного логічного блока, так і декількох. Логічний вираз, що містить декілька відношень, записується за допомогою логічних операцій.

Наприклад, A<X<B можна записати як X>A ∧ X<B.

На рисунку 7 наведено схему алгоритму обчислення функції Z, в якому логічні вирази поділені на відношення, що їх складають. Блоки 3, 4, 5 реалізують логічне множення, блоки 6, 7 – логічне додавання.

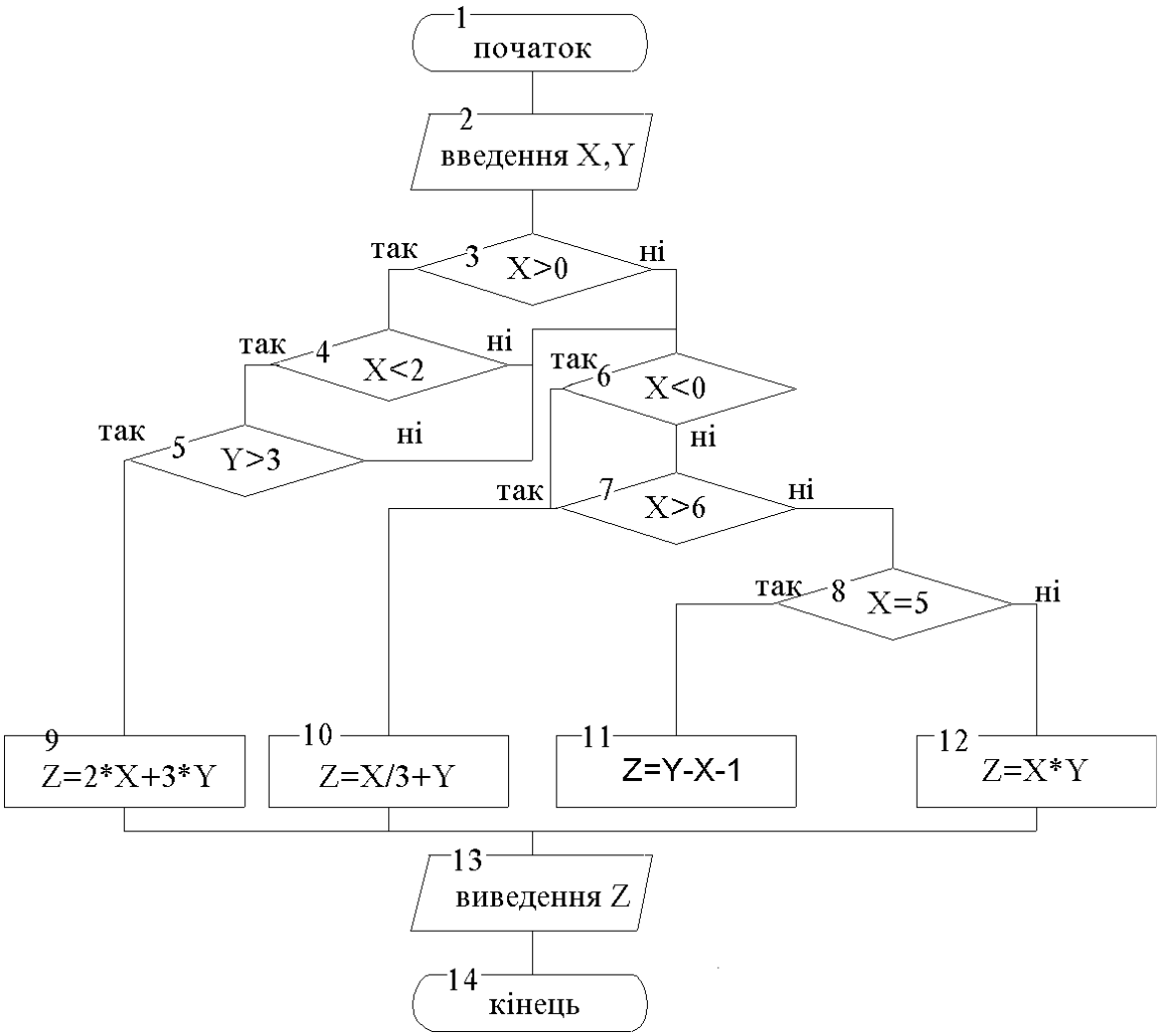


Рисунок 7 – Алгоритм обчислення функції Z

Контрольні запитання

1 Поняття розгалуженого обчислювального процесу.

2 Поняття логічного відношення.

3 Поняття логічного виразу.

4 Логічні операції.

5 Який блок використовується для задання управляючої частини алгоритму?

3 ПРОСТІ АРИФМЕТИЧНІ ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ

У практиці інженерних розрахунків доводиться виконувати багаторазове обчислення за однаковими математичними формулами при різних значеннях вхідних даних. Такі обчислення називають **циклами.**

Циклічним називається алгоритм, який містить послідовність операцій, що виконуються багаторазово. Використання циклів дозволяє значно зменшити схему алгоритму та розмір відповідної програми.

За способом задання числа повторень тілу циклу можна виділити арифметичні та ітераційні цикли.

Арифметичними називають цикли, число повторень яких задається в умові задачі або, за необхідності, розраховується до початку виконання циклу. В таких циклах наявна змінна, яка називається **параметром циклу**. Параметр циклу має бути визначеним межами та кроком його зміни, які задаються константами, змінними або виразами. Арифметичні цикли з одним параметром називаються **простими**. Умова закінчення арифметичного циклу може перевірятися у різних місцях алгоритму.

В **ітераційних циклах** кількість повторень, як правило, не може бути визначена до початку виконання циклу і залежить від заданої умови виходу з нього.

Рисунок 8 – Цикл з постумовою

Рисунок 9 – Цикл з передумовою

Цикл з постумовою (ЦИКЛ-ДО) (рисунок 8). Перевірка умови закінчення циклу здійснюється у кінці циклу після виконання дій (тіла циклу). У зв'язку з цим тіло циклу завжди виконується хоча б один раз.

Цикл з передумовою (ЦИКЛ-ПОКИ) (рисунок 9). Умова перевіряється на початку циклу до виконання тіла циклу, тому тіло циклу може взагалі не виконуватися.

Приклад 4. Скласти схему алгоритму обчислення значень функції

$$Y = \sin(XA) - B.$$

Параметр циклу X змінюється в інтервалі від $X_П$ до $X_К$ з кроком H_X, де $X_П = C$, $X_К = D$, $Hx = H$, $A = 9.63$, $B=5.1$; C, D, H – довільні числа (C<D).

Схема алгоритму задачі являє собою простий циклічний обчислювальний процес з неявно заданим числом повторень.

У схемі алгоритму, що наведена на рисунку 10, цикл організований з використанням блока 5 «модифікатор». У цьому блоці визначені початкове, кінцеве значення параметра циклу X та крок його зміни. Обчислення значення функції виконується у блоці 6.

На рисунку 11 наведена схема алгоритму з блоком «рішення». Її блоки 5, 6, та 9 відповідають блоку 5 схеми рисунка 10. Призначення блоків:

блоки 2 – 4 – формування вхідних даних;

блок 5 – задання початкового значення параметру X;

блок 6 – перевірка умови закінчення циклу;

блок 7 – обчислення функції;

блок 8 – друкування значення функції та значення параметра циклу;

блок 9 – обчислення наступного значення параметра циклу.

Рекурентний вираз – це вираз, який визначає своє значення через звернення до себе самого з іншими аргументами. Обчислення за рекурентними виразами реалізується циклічними обчислювальними процесами.

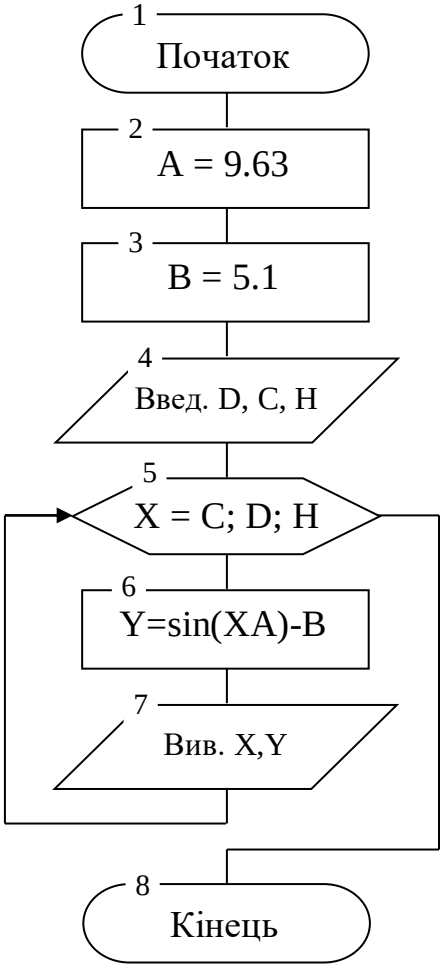


Рисунок 10 – Цикл з використанням блока «модифікатор»

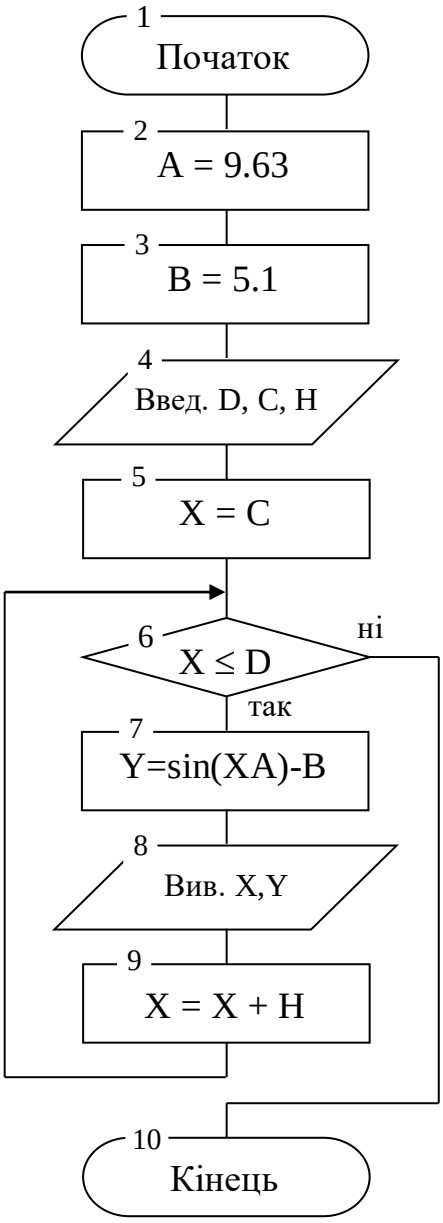


Рисунок 11 – Цикл з використанням блока «рішення»

Наприклад, формула X=X+1 означає: до вмісту X додати 1 та результат записати у X, тобто рекурентний вираз пов’язує між собою послідовно обчислені значення. Вхідними даними для наступного кроку є результати попереднього.

Рекурентні вирази використовуються для обчислення суми або добутку кінцевого числа даних. Для того щоб обчислити суму деякого числа даних, необхідно виконати дії:

- сформувати початкові дані;

- визначити початкове значення змінної, в якій буде здійснюватися накопичування суми;

- організувати цикл накопичування суми шляхом додавання нових значень до суми всіх попередніх;

- вивести результат.

Приклад 5. Скласти схему алгоритму обчислення значення факторіалу Y=N!

Факторіал – це добуток чисел натурального ряду від 1 до N (при N>0). Алгоритм визначення факторіалу наведено на рисунку 12.

Призначення блоків:

блок 2 – введення N. N – число, факторіал якого визначається;

блок 3 – присвоєння початкового значення змінній Y, в якій буде накопичуватися добуток. Для зберігання добутку початкове значення змінної Y дорівнюватиме 1.

блоки 4 – 7 – організація циклу для накопичування добутку;

блок 8 – виведення результату.

Приклад 6. Скласти схему алгоритму обчислення суми 15 значень функції: Z=sin(AX+B). Початкове значення X=2; крок зміни H=1.5; A, B – довільні числа.

Алгоритм розв’язання наведено на рисунку 13.

Призначення блоків:

блоки 2–3 – формування вхідних даних;

блок 4 – присвоєння початкового значення змінній S;

блоки 5–8 – організація циклу з параметром N; розрахунок значень функції Z; додавання отриманих значень; збільшення X на розмір кроку;

блок 9 – друкування значення суми.

Контрольні запитання

1 Який алгоритм називається циклічним?

2 Поняття арифметичного циклу.

3 Поняття ітераційного циклу.

4 Які цикли називаються простими?

5 Який цикл називають циклом з постумовою?

6 Який цикл називають циклом з передумовою?

7 Якими трьома блоками можна замінити блок «модифікатор»?

8 Що таке рекурентний вираз?

9 Алгоритм обчислення значення суми.

10 Алгоритм обчислення значення добутку.

11 Алгоритм обчислення значення факторіалу.

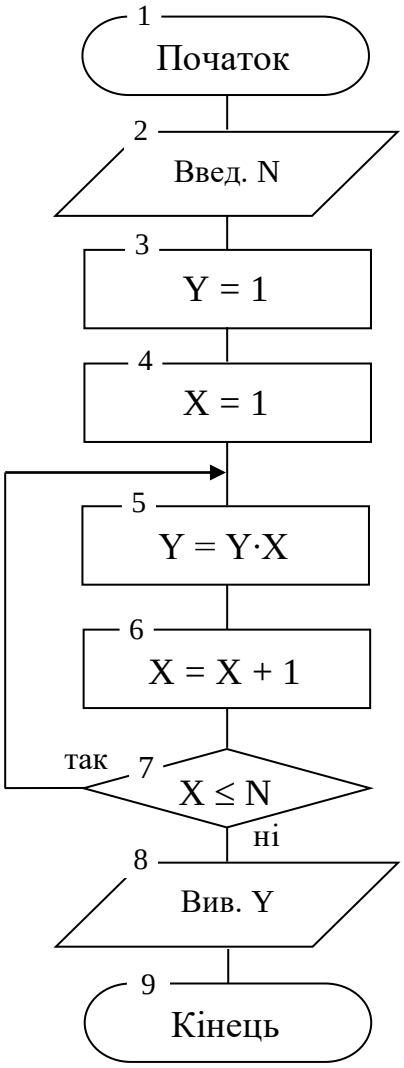


Рисунок 12 – Алгоритм обчислення значення факторіалу

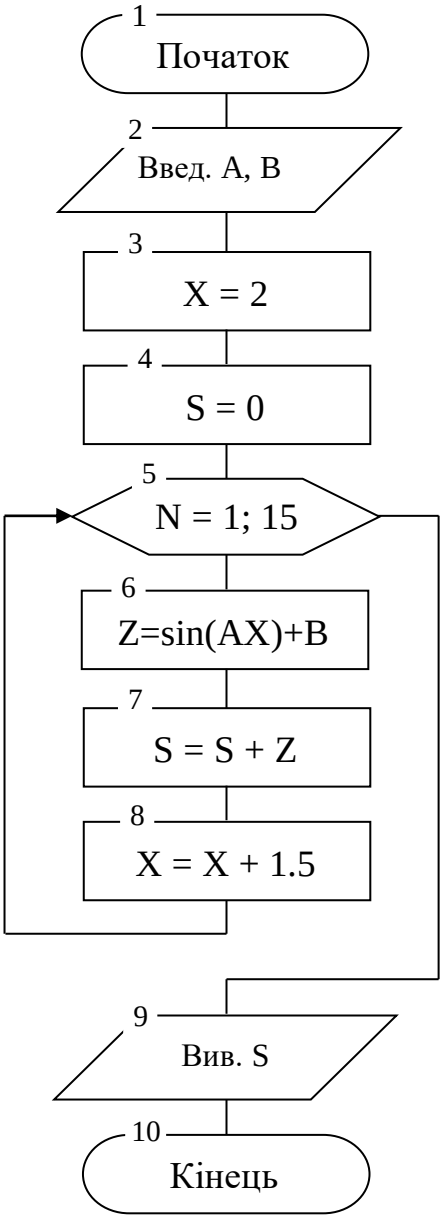


Рисунок 13 – Алгоритм обчислення суми 15 значень функції

4 ІТЕРАЦІЙНІ ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ

Поряд з арифметичними циклами в інженерній практиці використовуються ітераційні циклічні обчислювальні процеси.

Ітераційний цикл – це циклічний обчислювальний процес, в якому число повторень тіла циклу заздалегідь невідоме і залежить від умови досягнення шуканого результату. В ітераційних алгоритмах необхідно забезпечити обов'язкове виконання умови виходу з циклу, тобто збіжність ітераційного процесу.

Приклад 7. Скласти схему алгоритму обчислення значення функції Y, що подана сумою елементів нескінченного числового ряду, що сходиться, вигляду:

$$Y = \frac{1}{1+a} + \frac{2}{1+a^2} + \frac{3}{1+a^3} + \frac{4}{1+a^4} + \cdots + \frac{i}{1+a^i} + \cdots$$

для значення a >1.0.

Так як числовий ряд нескінченний, то для практичних розрахунків обмежуються деяким числом елементів виходячи з вимоги заданої точності E обчислення суми Y.

Числовий ряд, що сходиться – це ряд величин (елементів ряду), значення кожної з яких менше значення попередньої величини цього ряду. Сума елементів такого ряду – це скінченна величина і її можна знайти. Практично обчислення суми елементів припиняють на черговому елементі, що за своїм значенням менше, ніж задана точність E. Усіма наступними елементами (які виявляються меншими за точність E) нехтуємо. Якщо абсолютні значення елементів змінюються так, як показано на рисунку 14, то в суму будуть включені тільки чотири перших елементи. П’ятий елемент і наступні за ним у суму не включаються .

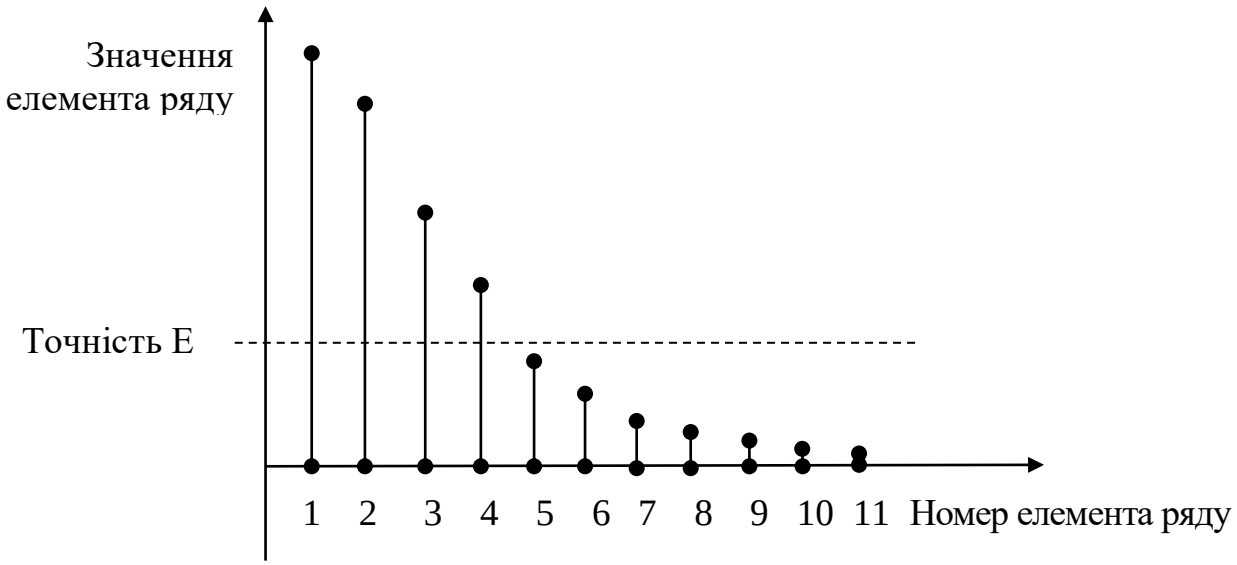


Рисунок 14 – Абсолютні значення елементів

На рисунку 15 показано алгоритм розрахунку суми Y. Накопичування суми елементів виконується в блоці 7 за допомогою рекурсивного виразу Y = Y+Z, де Z – значення чергового обчисленого доданка суми.

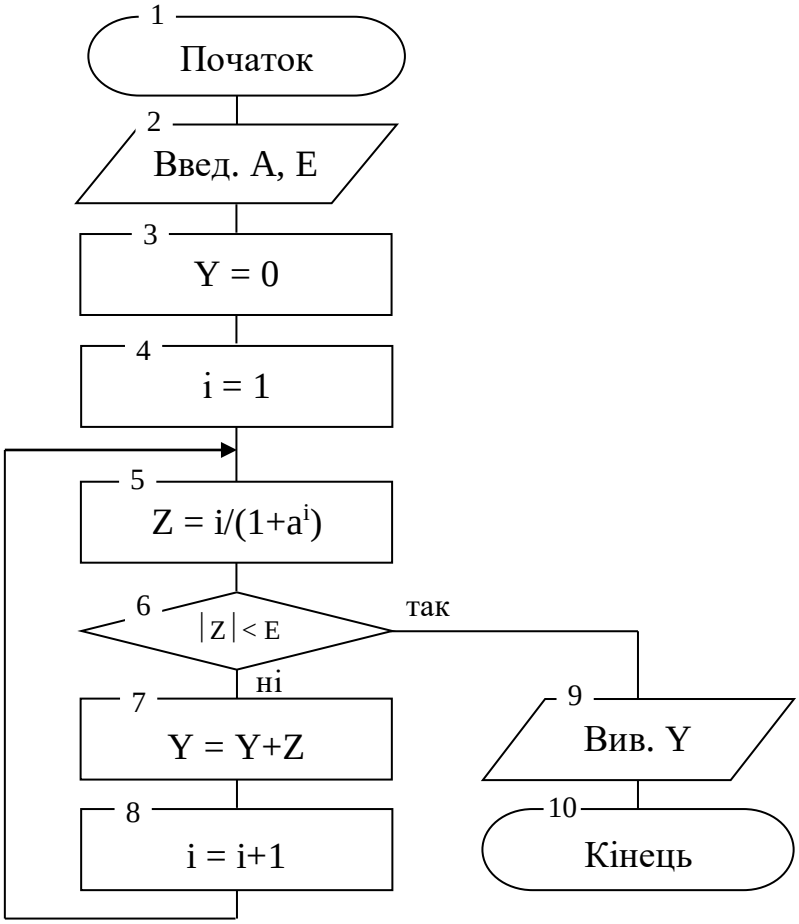


Рисунок 15 – Алгоритм ітераційного обчислювального процесу

В наведеному алгоритмі блок 6 – перевірка: чи слід даний доданок Z додавати в суму Y? Це необхідно робити, якщо абсолютне значення доданка $|Z| > E$ і його значенням не можна нехтувати. Перевірка рядів, у яких можуть існувати від'ємні елементи, також виконується за модулем значення елемента. Блоки 3 і 4 задають початкові значення суми Y і допоміжної величини **i**.

При побудові ітераційного обчислювального процесу неприпустимо використання блоку «модифікатор», тому що в ітераційному процесі немає чітко визначеного кінця циклу (кількості повторень).

Вихід з ітераційного циклу може бути виконаний на будь-якому кроці при досягненні заданої умови точності. Мається на увазі, що якщо в наведеному прикладі значення величини **a** буде вводитися різним, то різним буде і значення чергового доданка Z. Отже, число повторень за однієї і тієї самої точності спрогнозувати заздалегідь неможливо.

Приклад 8. Скласти схему алгоритму обчислення суми елементів ряду Y з точністю $E = 10^{-5}$.

$$Y = B + \frac{A}{2X+1} - \frac{A^2}{6X+2} + \frac{A^3}{24X+3} - \frac{A^4}{120X+4} + \cdots + \frac{(-1)^i A^{i-1}}{i! \cdot X + (i-1)},$$

вивести на друк значення всіх елементів, що входять у суму і визначити їх кількість.

Алгоритм обчислення суми елементів ряду наведено на рисунку 16. Допоміжна змінна **i** вказує на порядковий номер поточного елемента. Останнє значення **i** буде кількістю елементів ряду.

Контрольні запитання

1 Який цикл називають ітераційним?

2 З якою метою задається точність обчислення суми?

3 Чи допускається в ітераційному обчислювальному процесі використання блоку «модифікатор»?

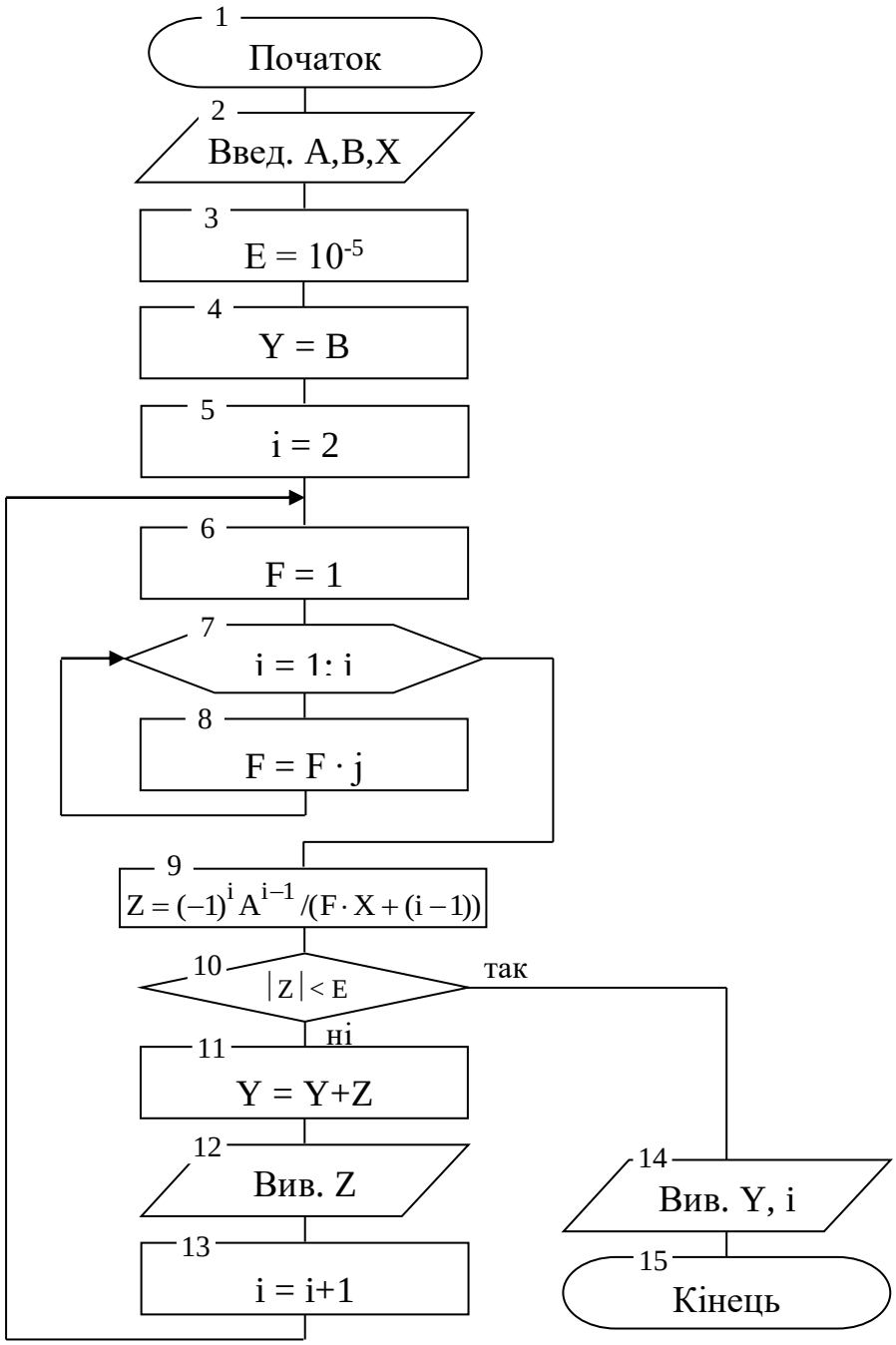


Рисунок 16 – Алгоритм обчислення суми елементів ряду до прикладу 8

5 ВКЛАДЕНІ ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ

Поряд із простими циклічними процесами при побудові алгоритмів складних перетворень використовуються вкладені циклічні обчислювальні процеси. Вкладені цикли застосовуються

у випадку розрахунків функцій, що залежать від декількох аргументів.

Вкладеним називається цикл, що містить в собі один чи декілька інших циклів. Цикл, що охоплює інші цикли, називається **зовнішнім**, а інші – **внутрішніми**. Основне правило побудови вкладених циклів – це охоплення зовнішнім циклом внутрішнього чи декількох внутрішніх. Кожен цикл окремо організується точно так, як і простий цикл.

Приклад 9. Скласти схему алгоритму обчислення значення функції $y = a\sin(x + a)$, якщо $x \in [0; 1.7]$, $h_x = 0.1$ і $a \in [0; 2]$, $h_a = 0.2$.

Параметри циклів змінюються послідовно, тобто для одного значення параметра зовнішнього циклу параметр внутрішнього циклу набуває послідовно усіх своїх значень (рисунок 17).

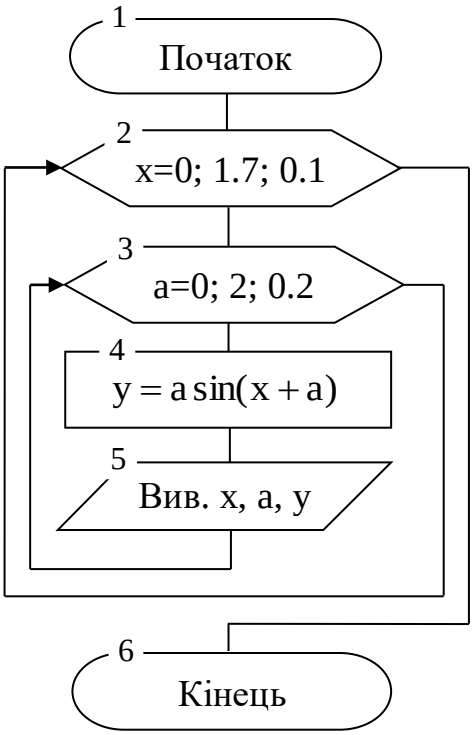


Рисунок 17 – Алгоритм обчислення значень функції двох аргументів

Загальна кількість обчислень тіла циклу визначається добутком числа повторень зовнішнього циклу і внутрішнього циклу.

Вкладеними в циклічних процесах можуть бути не тільки інші цикли, але і розгалуження і фрагменти лінійного типу.

Приклад 10. Скласти схему алгоритму обчислення і виведення на друк значень функції

$$y = \begin{cases} a\sin(x + a), & \text{якщо } x < 0 \text{ і } a < x; \\ -b - \cos(x - a), & \text{в інших випадках,} \end{cases}$$

$$x \in [-3; 3], \quad h_x = 0.1, \qquad a \in [-10; 10], \quad h_a = 0.5.$$

В алгоритмі, показаному на рисунку 18, у внутрішньому циклі наявний розгалужений обчислювальний процес. Кожна гілка розгалуження може містити фрагменти лінійного типу (блоки 7, 8).

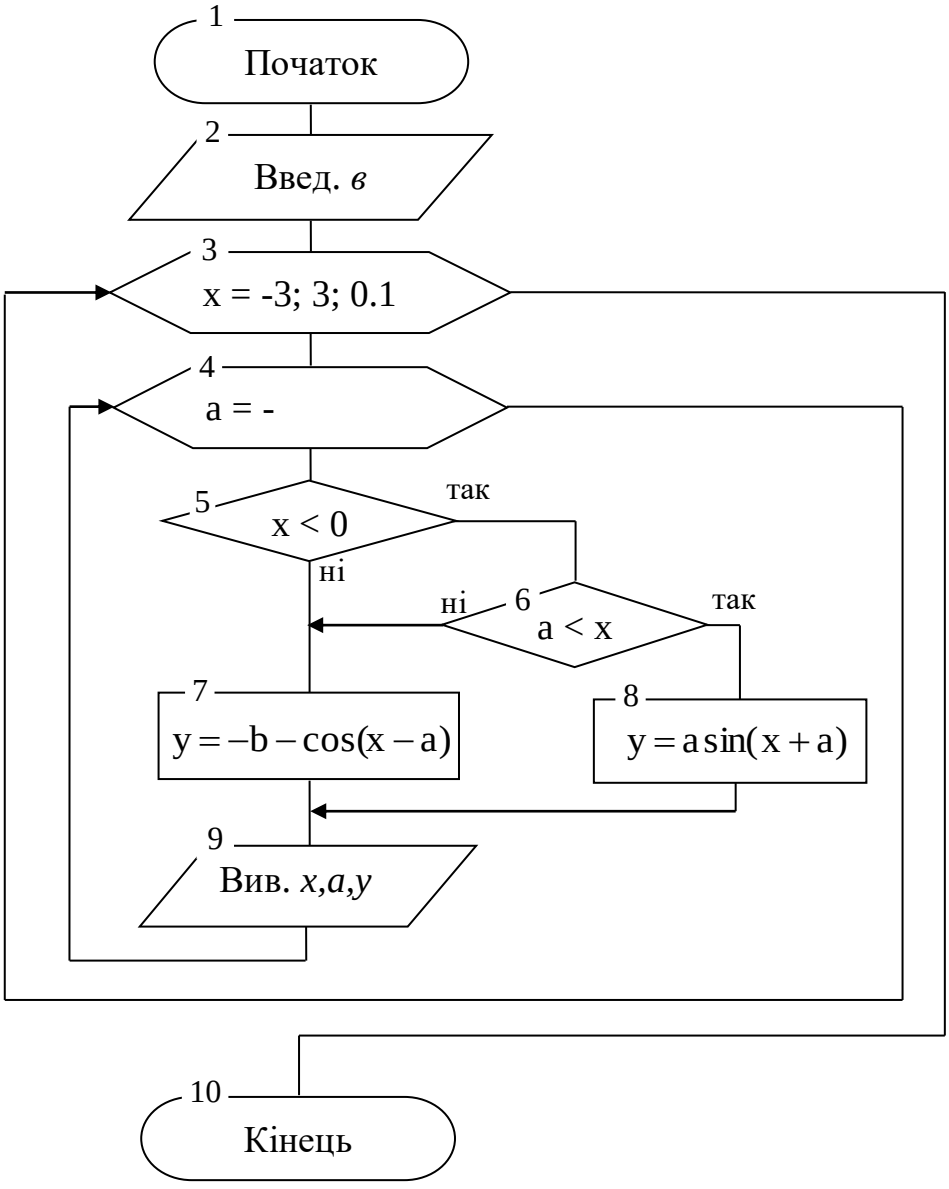


Рисунок 18 – Алгоритм вкладеного циклічного процесу з розгалуженнями

Контрольні запитання

1 Який алгоритм називається циклічним?
2 Який цикл називається вкладеним?
3 Який цикл називається зовнішнім?
4 Який цикл називається внутрішнім?

6 ВИЗНАЧЕННЯ НАЙБІЛЬШОГО ТА НАЙМЕНШОГО ЗНАЧЕНЬ ФУНКЦІЇ

Знаходження MIN та MAX значення функції припускає, що функція визначена на заданому інтервалі та необхідно визначити найбільше (MAX) чи найменше (MIN) серед них. Оскільки аналітичний вираз функції незмінний на інтервалі визначення, то різні її значення утворюються за рахунок зміни параметрів (аргументів) цієї функції. Тому процедура визначення MIN та MAX значення функції виконується у циклі.

Перед початком циклу з визначення MIN чи MAX значення функції Y=f(x) обчислюється перше значення Y1 функції, яке приймається за MIN чи MAX, тобто Ymin=Y1 чи Ymax=Y1. Далі для обчислення наступного значення функції змінюємо значення параметра циклу та, якщо нове значення параметра припустиме, обчислюємо відповідне значення функції. Отримане значення функції Y2, в залежності від умови задачі, порівнюємо при знаходженні MIN з Ymin, при знаходженні MAX з Ymax.

Якщо Y2< Ymin, то Ymin набуває значення Y2 (Ymin=Y2), а якщо Y2>=Ymin, то Ymin залишає своє значення, після чого здійснюється перехід до обчислення наступного значення параметра. Після зміни параметра циклу процес повторюється.

При знаходженні MAX, Y2 порівнюється з Ymax. Якщо Y2>MAX, то Ymax набуває значення Y2 (Ymax=Y2), інакше Ymax зберігає своє значення і після зміни параметра циклу процес повторюється.

Все це можна описати математичною залежністю

$$\text{MAX}: \quad Y\max = \begin{cases} Yi \quad , & \text{якщо } Yi > Y\max \\ Y\max, & \text{якщо } Yi \le Y\max \end{cases};$$

$$\text{MIN}: \quad Y\min = \begin{cases} Yi \quad , & \text{якщо } Yi < Y\min \\ Y\min \quad , & \text{якщо } Yi \ge Y\min \end{cases}.$$

де Yi – наступне значення функції.

Вихід з циклу здійснюється після досягнення параметром верхньої межі інтервалу. Слід зазначити, що при розв'язанні таких задач йдеться не про MIN чи MAX функції, а про MIN чи MAX серед обчислених значень цієї функції. Це пояснюється тим, що комп'ютер обчислює дискретні значення функції при відповідних дискретних значеннях параметра та дійсний MIN чи MAX може бути між ними (рисунок 19). Підвищити точність визначення MIN чи MAX можна за рахунок зменшення кроку зміни параметра циклу.

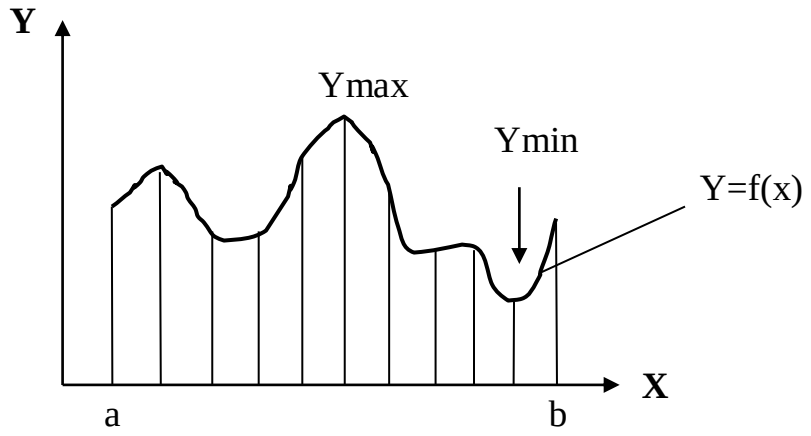


Рисунок 19 – Максимальне та мінімальне значення функції.

Приклад 11. Скласти схему алгоритму визначення мінімального значення функції $y = |a| \cdot e^{bx}$, при зміні параметра x від 0 до 4 з кроком h=0.5.

Алгоритм пошуку MIN значення цієї функції наведено на рисунку 20. Призначення блоків:

блок 2 – введення змінних a, b;

блок 3 – надання початкового значення параметру циклу x;

блок 4 – обчислення першого значення функції;

блок 5 – змінній MIN присвоюється перше значення функції Y;

блок 6 – блок управління циклом;

блок 7 – обчислення наступного значення функції;

блок 8 – порівняння наступного значення функції з MIN. Якщо чергове значення функції менше ніж MIN, то MIN набуває цього значення (блок 9) та здійснює перехід до чергового значення параметра. У протилежному випадку MIN зберігає своє значення та здійснює перехід до чергового значення параметра;

блок 10 – друк MIN значення функції;

блок 11 – вихід з алгоритму.

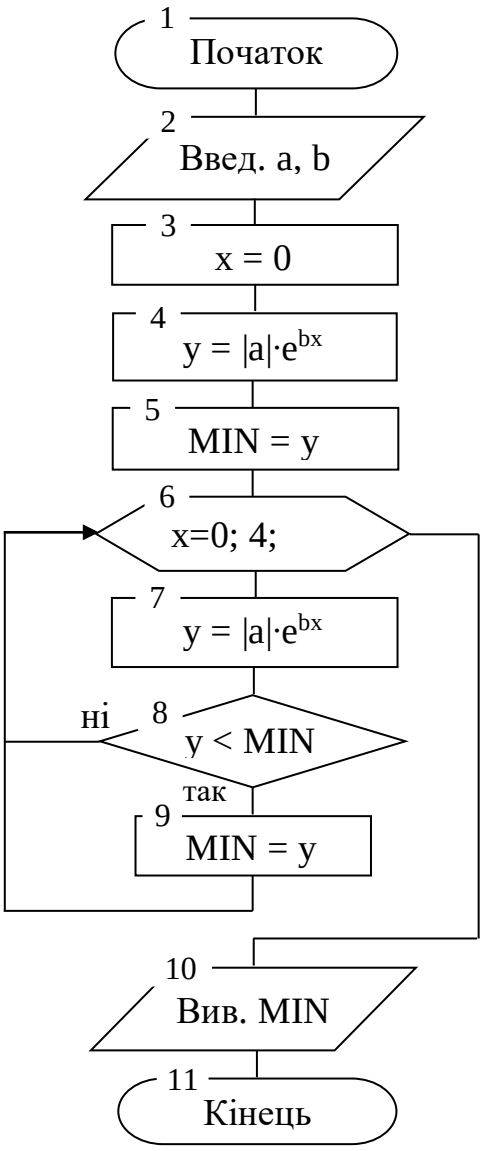


Рисунок 20 – Алгоритм визначення мінімального значення функції

Контрольні запитання

1 Що таке екстремуми функції?

2 Який основний обчислювальний процес використовується для визначення екстремумів функцій?

3 Чи може бути на заданому інтервалі декілька екстремумів функції?

7 ПОШУК І ВИБІРКА ЕЛЕМЕНТІВ МАСИВУ

Масив – упорядкована сукупність елементів одного типу, що має ім'я. Позиція елемента в масиві може визначатися декількома вимірами і вказується індексами, що записуються з ім'ям масиву.

Індекс – константа, змінна чи вираз.

Елементом масиву називають змінну з індексом. Значення змінним з індексом присвоюють за такими ж правилами, що і простим змінним.

Масиви характеризуються розміром та розмірністю. **Розмір масиву –** кількість елементів масиву. **Розмірність масиву** – кількість вимірювань масиву (кількість індексів).

$A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, ..., a_n\}$ – одновимірний масив.

$$A = \begin{Bmatrix} a_{11},\ a_{12},\ a_{13},\ ...,\ a_{1n} \\ a_{21},\ a_{22},\ a_{23},\ ...,\ a_{2n} \\ ... \\ a_{m1}, a_{m2}, a_{m3}, ..., a_{mn} \end{Bmatrix}$$ – двовимірний масив.

Алгоритмами пошуку і вибірки реалізують процедури вибірки одиночних елементів чи елементів групи, формування масивів елементів, значення яких задовольняють визначені умови (ключові ознаки). Умови вибірки задаються виразами арифметичного або логічного типів.

Ключова ознака – одна чи декілька характеристик, що відрізняють один елемент від іншого (порядковий номер елемента в масиві, значення самого елемента і т.д.)

Приклад 12. Масив X складається з N числових констант X_i. Розробити схему алгоритму формування масиву B, що складається з додатних чисел масиву X.

В алгоритмі (рисунок 21) перевірку відповідності ознаки виконує блок 6. Блок 8 формує нову послідовність елементів, що утворюють масив B з елементами B_j. Кінцеве значення змінної j відповідає розміру масиву B.

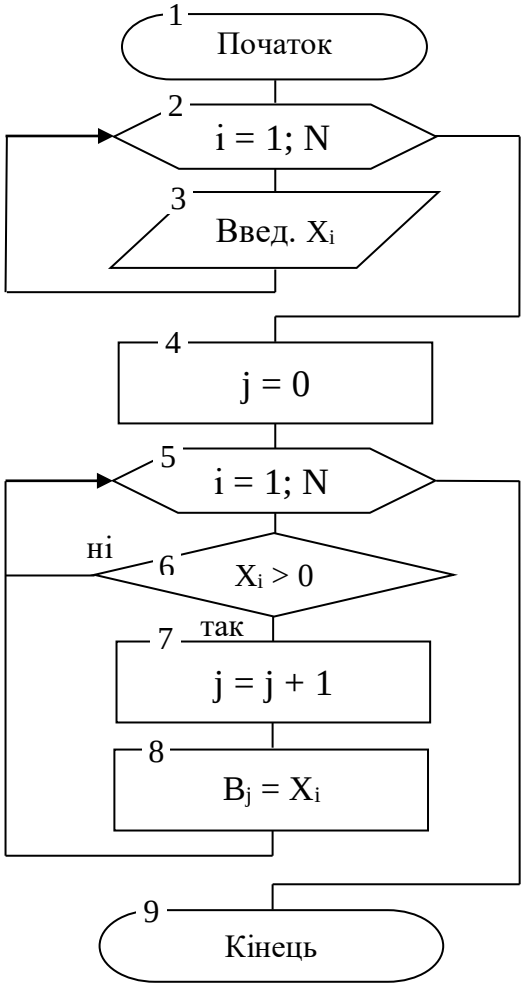


Рисунок 21 – Алгоритм формування масиву B, що складається з додатних чисел масиву X

Приклад 13. Масив X складається з N числових констант X_i. Розробити схему алгоритму визначення елемента X_i з найбільшим (MAX) значенням.

В алгоритмі (рисунок 22) при визначенні елемента з максимальним значенням значення кожного елемента масиву порівнюються із змінною MAX (блок 6), якій у блоці 4 присвоюється значення першого елемента масиву. Блок 7 змінює

максимальне значення елементів масиву в змінній MAX, якщо наступний елемент X_i > MAX.

Після перебору всіх елементів масиву значення змінної MAX буде відповідати найбільшому значенню елементів масиву X.

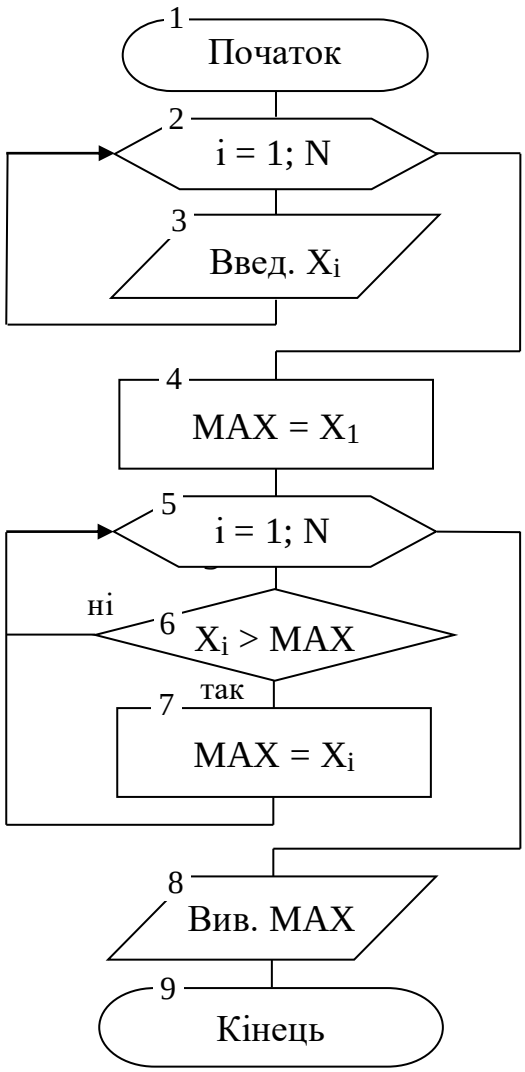


Рисунок 22 – Алгоритм визначення максимального значення елементів масиву

Контрольні запитання

1 Що таке масив?

2 Характеристики масиву.

3 Що таке ключова ознака?

4 Який основний тип обчислювальних процесів використовується для обробки масивів?

5 Чи можна в одному масиві зберігати цілі та дійсні числа?

8 ОБРОБКА ОДНОВИМІРНИХ МАСИВІВ

Схеми алгоритмів обчислення суми і добутку кінцевого числа елементів масиву являють собою циклічні алгоритми. Параметр циклу – порядковий номер елемента масиву.

Алгоритм, що наведений на рисунку 23, дозволяє визначити середнє арифметичне значення 20 елементів масиву K, i – порядковий номер елемента масиву.

У блоці 6 накопичується сума S при послідовному переборі елементів масиву. Початковий стан S – нуль, що визначається блоком 4. Середнє арифметичне значення SR кінцевого числа елементів масиву визначається за формулою $SR = \frac{S}{K}$, де S – сума елементів, K – кількість елементів.

Схема алгоритму, що наведена на рисунку 24, дозволяє визначити добуток 20 елементів масиву K. У блоці 6 накопичується добуток P при послідовному переборі елементів масиву. Початковий стан P – одиниця, що визначається блоком 4.

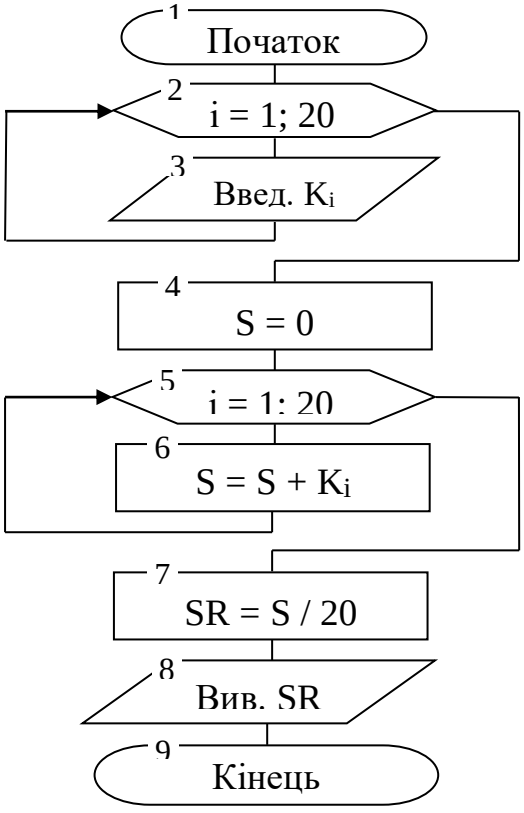


Рисунок 23 – Алгоритм визначення середнього арифметичного значення елементів масиву

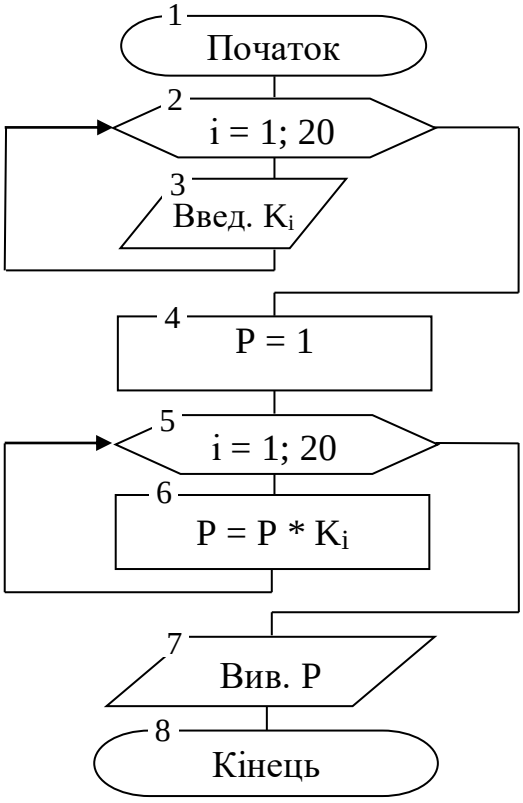


Рисунок 24 – Алгоритм визначення добутку елементів масиву

9 ОБРОБКА ДВОВИМІРНИХ МАСИВІВ

Для двовимірних масивів процедури введення, обробки, виведення елементів будуються на основі вкладених циклічних обчислювальних алгоритмів.

Принципи побудови алгоритмів обробки двовимірних масивів розберемо на прикладі розв’язання задачі.

Приклад 14. Задано двовимірний масив X числових елементів (K рядків і K стовпців). Визначити:

– різницю між добутком і сумою елементів > 10 у масиві X;

– максимальний елемент головної діагоналі масиву X.

Схему алгоритму наведено на рисунку 25.

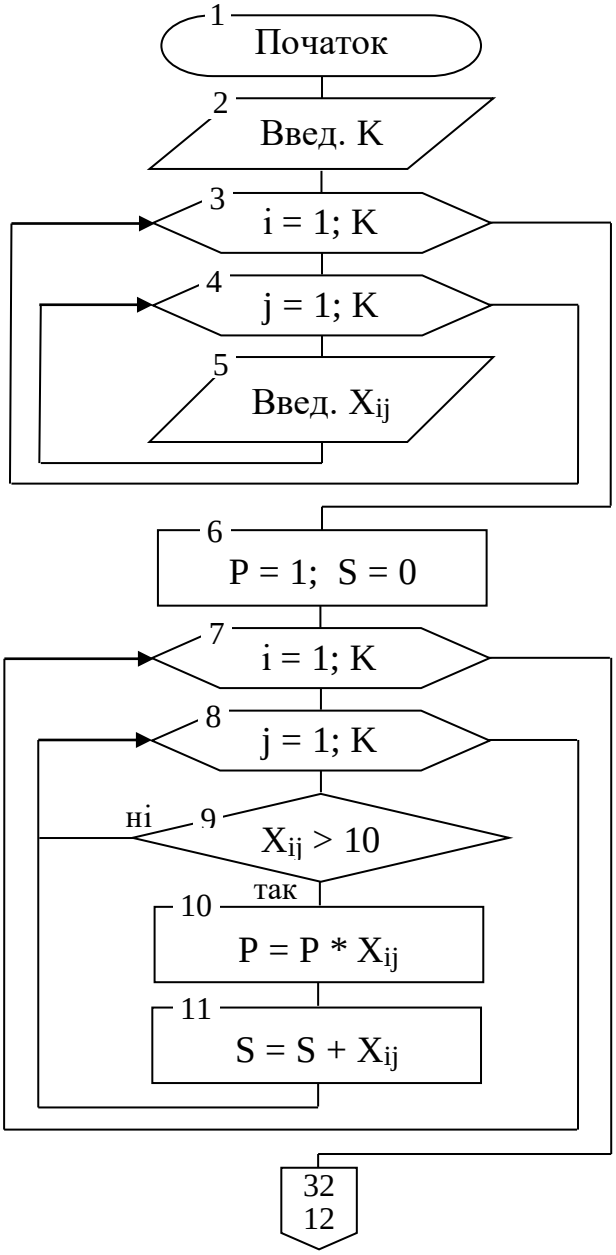


Рисунок 25 – Алгоритм до прикладу 14

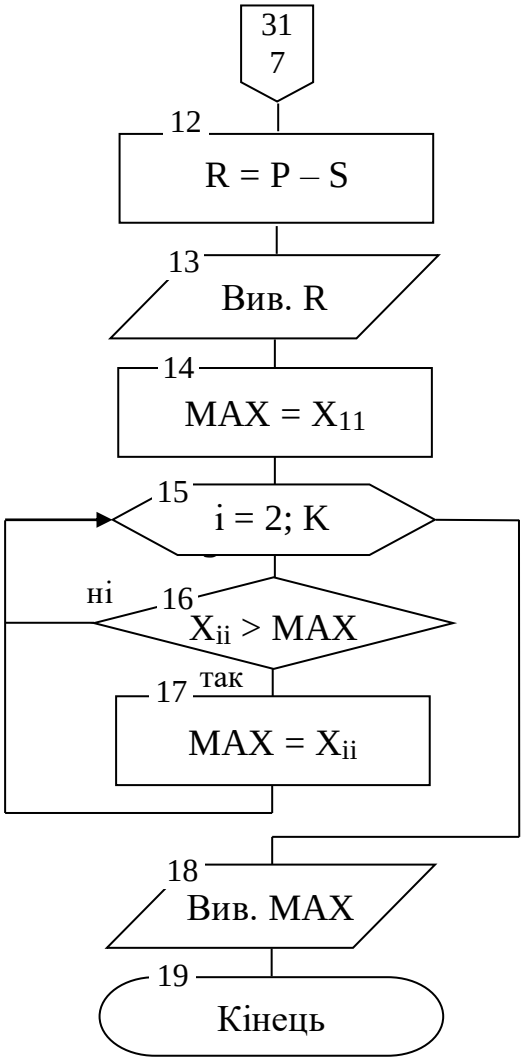


Рисунок 25, аркуш 2

10 ПРОЕКТУВАННЯ АЛГОРИТМІВ СОРТУВАННЯ

Алгоритми сортування змінюють відносне розташування елементів оброблюваного масиву без зміни їхнього значення. Найбільш загальним видом сортування можна вважати упорядкування масиву (розташування елементів у порядку збільшення або зменшення значення ключової ознаки).

Організація сортування залежить від розміру масиву, виду пам'яті, використовуваної для його збереження. Умовно масив вважається невеликим, якщо його розмір не перевищує ємності оперативної пам'яті, виділеної для роботи з ним. Оперативна пам'ять допускає звернення до будь-якого елемента масиву

шляхом безпосереднього задання адреси (пам'ять з довільним доступом). Масиви невеликих обсягів сортуються методами внутрішнього сортування.

Для збереження великих масивів, розмір яких істотно перевищує ємність оперативної пам'яті, використовується зовнішня пам'ять (магнітна стрічка, магнітні диски). Такі масиви сортуються методами зовнішнього сортування.

Алгоритм вставки (рисунок 1). Алгоритм вставки виконує «вставку» кожного елемента в таке місце масиву, щоб при переході від елемента до елемента ключова ознака X(i) змінювалася монотонно. Алгоритм застосовується для сортування масивів, збережених у пам'яті з довільним доступом.

Приклад 15. Елементи масиву X нумеруються в пам'яті в порядку їхнього розташування числами натурального ряду i=1,2,...,N. Скласти схему алгоритму сортування масиву в порядку збільшення значення ключової ознаки.

Схему алгоритму наведено на рисунку 26. X(i) - ключова ознака елемента **i** (його числове значення).

Ознака кожного елемента, починаючи з i=2, послідовно порівнюється з ознакою попередніх елементів j=i-1, i-2,...,1 (блок 5). Значення ознаки елемента i попередньо переписується у додатковий елемент пам'яті S (блок 4).

Якщо X(i)<X(j) (чи S<X(j)), то j елемент записується на j+1 місце (блок 6).

Якщо X(i)>X(j), то на місце j+1 записується елемент i (блок 9), j-й елемент залишається на місці.

Після перегляду всіх елементів вони будуть розставлені так, що ключова ознака X(i) елемента з меншим номером буде менше ключової ознаки елемента **i**+1 і більше ключової ознаки елемента **i**-1.

Приклад 16. Скласти схему алгоритму сортування другого рядка двовимірного масиву Z. Елементи масиву Z(i) розташувати в порядку збільшення значення ключової ознаки.

Робота алгоритму (рисунок 27) починається з визначення номера рядка i=2 (блок 2). Цикл на основі блока 3 послідовно перебирає елементи Z(2,J), J=1,2,3,…N.

N-кількість елементів рядка.
Робота блоків 4, 5, 6, 7, 8, 9, 10 описана у прикладі 15.

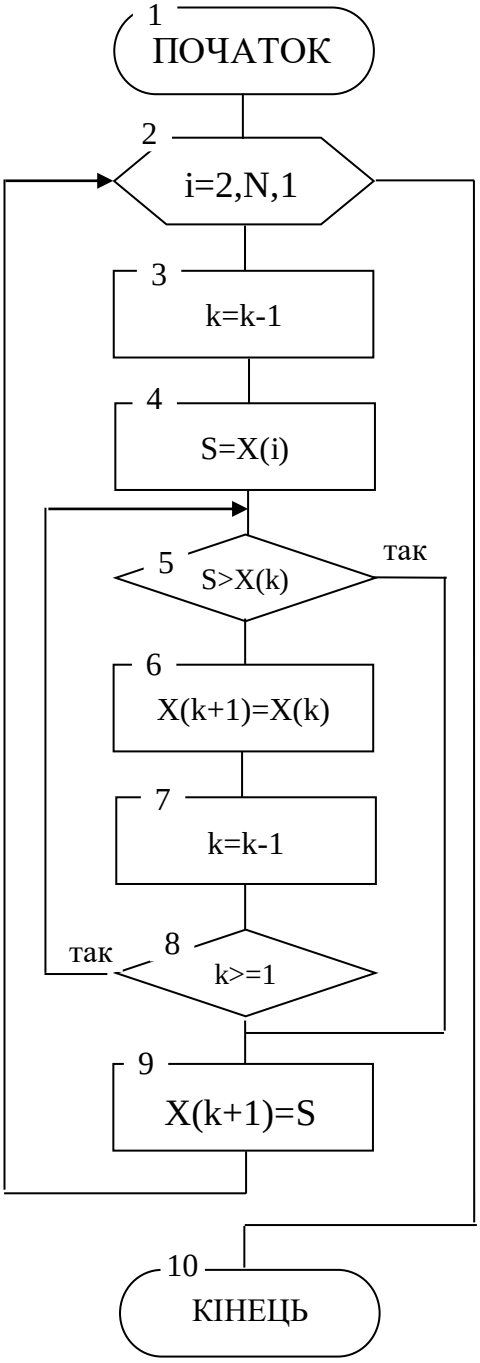


Рисунок 26 – Алгоритм вставки для одновимірних масивів (приклад 15)

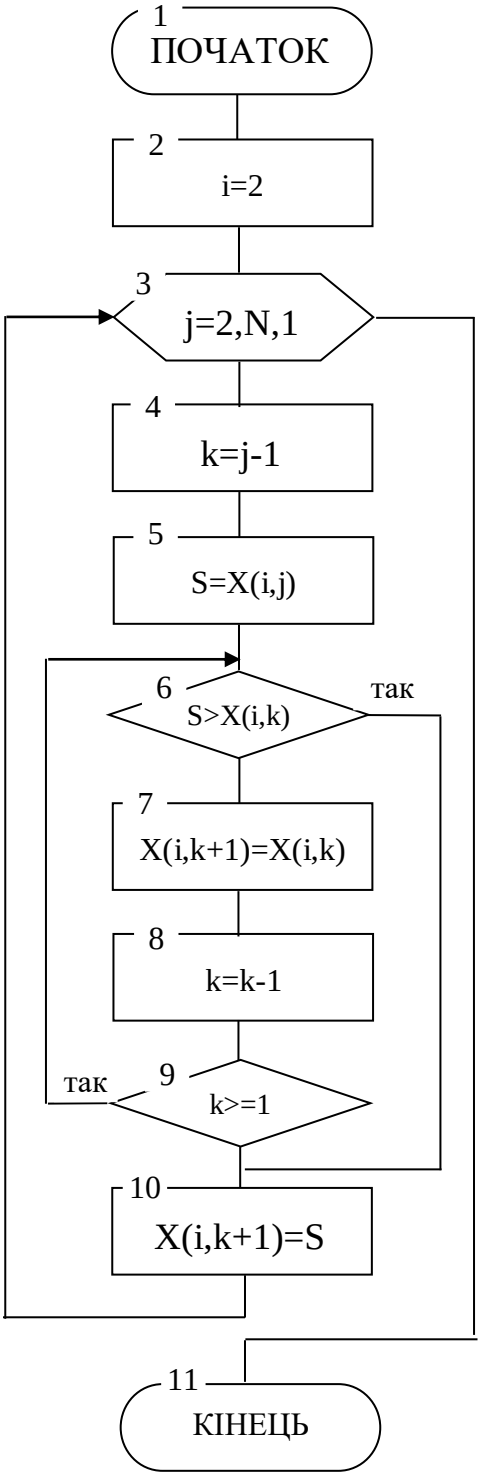


Рисунок 27 – Алгоритм вставки для двовимірних масивів (приклад 16)

СПИСОК ЛІТЕРАТУРИ

1 Вычислительная техника и программирование [Текст] : учеб. для техн. вузов / А. В. Петров, В. Е. Алексеев, А. С. Ваулин и др. – М. : Высш. шк., 1990. – 480 с.

2 Основи алгоритмізації базових обчислювальних процесів [Текст] : навч. посібник / О. В. Головко, В. С. Меркулов, В. О. Гончаров, І. Г. Бізюк, В. М. Бутенко. – Харків : УкрДАЗТ, 2008.– 163 с.

3 Алгоритмічні мови програмування [Текст] : метод. вказівки до лаб. робіт з дисц. «Комп'ютерна техніка та програмування». – Харків : УкрДАЗТ, 2012. – Ч. 1. – 30 с.

4 Основи проектування алгоритмів [Текст] : метод. вказівки до лаб робіт з дисц «Комп’ютерна техніка та програмування». – Харків : УкрДАЗТ, 2011. – 45 с.