МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

Г.Г.ШВАЧИЧ, О.А. ГУЛЯЄВА, О.І. ОРЖЕХ

АЛГОРИТМІЗАЦІЯ У ПРИКЛАДАХ ТА ЗАВДАННЯХ

Дніпропетровськ НМетАУ 2010

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

Г.Г.ШВАЧИЧ, О.А. ГУЛЯЄВА, О.І. ОРЖЕХ

АЛГОРИТМІЗАЦІЯ У ПРИКЛАДАХ ТА ЗАВДАННЯХ

Затверджено на засіданні Вченої ради академії
як навчальний посібник

Дніпропетровськ НМетАУ 2010

УДК 004(075.8)

Швачич Г.Г., Гуляєва О.А., Оржех О.І. Алгоритмізація у прикладах та завданнях: Навч. посібник. − Дніпропетровськ: НМетАУ, 2010. – 44с.

Викладені основні питання теорії і практики створення алгоритмів. Наведені приклади створення алгоритмів різних структур. Подані завдання для самостійної роботи студентів.

Призначений для студентів напряму 050202 − автоматизація та комп'ютерні технології (АВ01), а також для студентів усіх форм навчання.

Друкується за авторською редакцією.

Відповідальний за випуск Г.Г. Швачич, канд. техн. наук, проф.

Рецензенти: Ю.Є. Чернявський, канд. фіз.-мат. наук, доц. (ДНУ)
Ю.Н. Головко, канд. фіз.-мат. наук, доц. (НГУ)

1. АЛГОРИТМИ. ОСНОВНІ ПОНЯТТЯ ТА ВЛАСТИВІСТІ

Поняття алгоритму

В теперішній час поняття алгоритму – одне з фундаментальних понять науки «інформатика». Алгоритм — це послідовність дій над заданими об’єктами, чітко та однозначно визначаюча обчислювальний процес.

Ефективним методом побудови алгоритмів є метод покрокової деталізації, при якому завдання розбивається на кілька простих підзадач (модулів), і для кожного модуля створюється свій власний алгоритм.

Здебільшого модуль реалізує певний процес обробки інформації і застосовується як для окремого використання, так і для включення модуля в інші алгоритми. Застосування модульності при створенні алгоритмів дозволяє розбити великі завдання на незалежні блоки (модулі), усуває повторення стандартних дій і значно прискорює процес відлагодження алгоритму в цілому.

Найчастіше головний модуль алгоритму містить декілька інших модулів, створених раніше. Використовуючи модулі як складові великої конструкції, можна створювати алгоритми будь-якого ступеня складності, і при цьому не втрачати контролю за функціюванням алгоритму всієї задачі.

Такий метод називається структурним проектуванням алгоритму, він є універсальним і може використовуватися як для обчислювальних процесів (так зване системне програмування), так і для процесів реального життя.

Властивості алгоритму

1. **Дискретність** – процес розв’язку розбивається на кроки. Кожен крок – це одна дія або підпорядкований алгоритм. Таким чином полегшується процес знайдення помилок і редагування алгоритму.
2. **Визначеність (точність)** – кожен крок алгоритму має бути однозначно описаною дією і не містити двозначностей.
3. **Зрозумілість** – усі дії, включені до алгоритму, мають бути у межах компетенції виконавця алгоритму.
4. **Універсальність (масовість)** – алгоритм має виконуватися при будь-яких значеннях вхідних даних та початкових умов.
5. **Скінченність** – алгоритм має бути реалізований за кінцеве число кроків і повинен використовувати кінцевий набор вхідних значень.
6. **Результативність** – алгоритм має привести до отримання результату.

Способи подання алгоритмів

Алгоритми можуть бути подані

- словесно (засобами природної мови у вигляді плану дій)
- графічно (у вигляді блок-схем)
- у вигляді програм, написаних певною мовою програмування.

Найчастіше алгоритми обчислювальних процесів подаються у вигляді блок-схем, де кожний крок алгоритму представлено спеціальним блоком, який показує дію, яку треба виконати.

Графічному опису передує, як правило, побудова математичної моделі – математичного опису алгоритму. Такий опис полягає у формалізованому (із застосуванням математичних символів) поданні всіх розглядуваних залежностей і методів відшукання значень вихідних даних на підставі вхідних.

Призначення блоків випливає з їхніх назв. Блоки поєднуються між собою лініями потоку. Природні напрями потоків зверху вниз і зліва направо. Якщо напрямок потоку інший то лінія повинна мати стрілку.

Рекомендується не перетинати лінії потоку, а використовувати поєднувач блоків. Для використання поєднувача блоки мають бути попередньо пронумеровані, а сам поєднувач має містити цифру – номер блоку, з яким відбувається поєднання або номер блоку, з якого відбувається поєднання. Інакше: при з'єднуванні блоків можливо використовувати спільний символ, який записується в поєднувач блоків.

Основні види блоків та їх призначення наведені у таблиці 1.1.

2. АЛГОРИТМИ ЛІНІЙНОЇ СТРУКТУРИ

Лінійна структура використовується в алгоритмах, де одна дія виконується слідом за іншою послідовно і при цьому жодна з дій не пропускається і не повторюється.

Розглянемо приклади алгоритмів лінійної структури.

Приклад 2.1. Обчислити висоти трикутника зі сторонами **a, b, c** за формулами

$$h_a = \frac{2}{a}\sqrt{p(p-a)(p-b)(p-c)}$$

$$h_b = \frac{2}{b}\sqrt{p(p-a)(p-b)(p-c)}$$

$$h_c = \frac{2}{c}\sqrt{p(p-a)(p-b)(p-c)},$$

де $\mathrm{p} = \dfrac{\mathrm{a}+\mathrm{b}+\mathrm{c}}{2}$

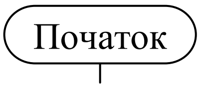
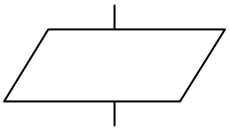
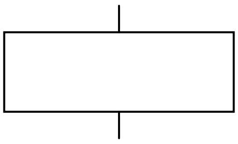
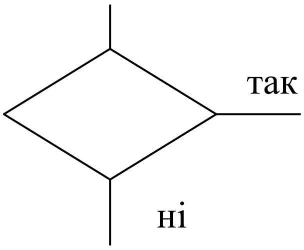
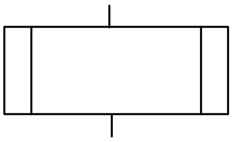
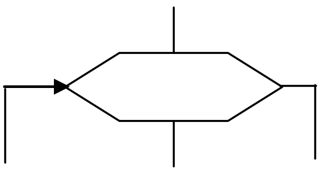
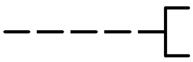
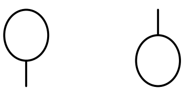
Введемо позначення:

$$t = 2\sqrt{p(p-a)(p-b)(p-c)} \Rightarrow h_a = \frac{t}{a}, \quad h_b = \frac{t}{b}, \quad h_c = \frac{t}{c}$$

Блок-схема алгоритму наведена на рисунку 2.1.

Таблиця 1.1.

Основні види блоків, їх призначення

Графічне зображення	Назва, виконувана дія
	Початок алгоритму
	Закінчення алгоритму
	Блок уводу-виводу
	Виконання обчислень або присвоєння значень
	Перевірка умови. Якщо умова справедлива (набуває значення ІСТИНА), виконується перехід по лінії **Так**, а якщо не справедлива (набуває значення БРЕХНЯ), то виконується перехід по лінії **Ні**
	Виклик раніше створених алгоритмів (модулів)
	Блок організації циклу
	Коментар. Короткі пояснення до показаного блоку
	Поєднувач блоків

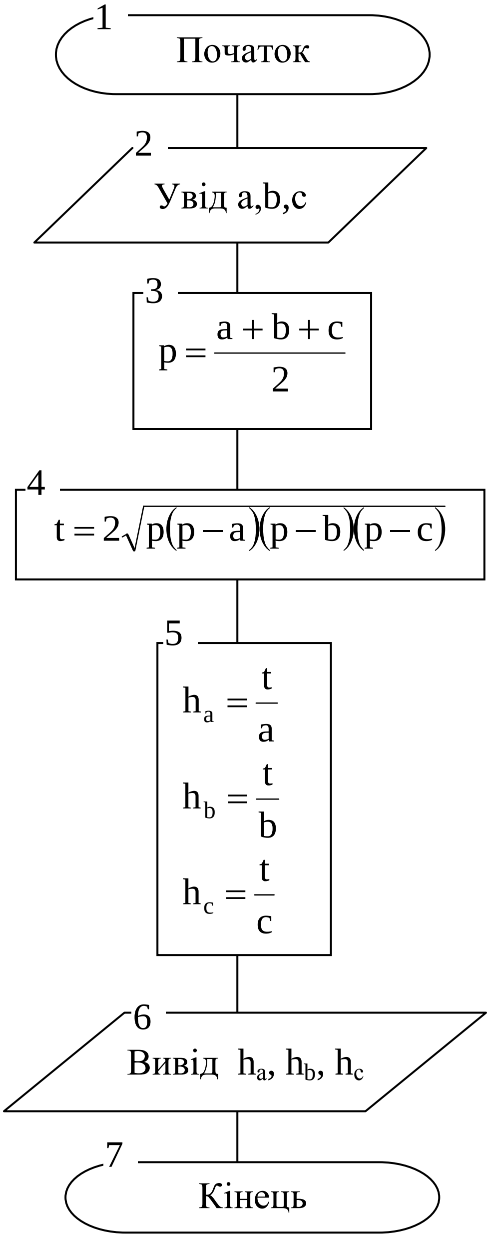


Рис. 2.1. Блок-схема алгоритму прикладу 2.1.

Приклад 2.2. Для заданих значень **a** і **c** знайти значення виразу:

$$Y = \sqrt{\left(\left|x^2 + c\right|\right)} - \sin(ax - c) \ ,$$

де $x = \cos(a^2c^2)$

Словесний спосіб надання алгоритму:

1. Уведення вхідних значень змінних **a, c**.
2. Обчислення виразу $x = \cos(a^2c^2)$.
3. Обчислення виразу $Y = \sqrt{\left(\left|x^2 + c\right|\right)} - \sin(ax - c)$.
4. Виведення вихідних значень **x** та **Y**.

На рисунку 2.2. подана блок-схема алгоритму.

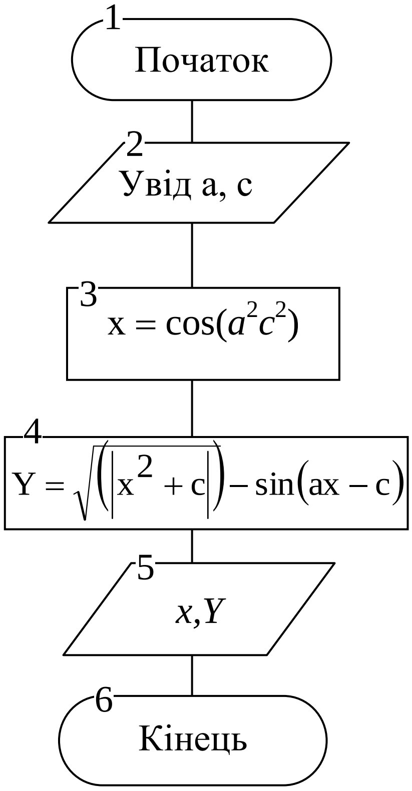


Рис. 2.2. Блок-схема алгоритму прикладу 2.2.

3. АЛГОРИТМИ РОЗГАЛУЖЕНОЇ СТРУКТУРИ

Розгалужена структура передбачає вибір виконання дії залежно від виконання певної умови, при цьому деякі дії можуть не виконуватися взагалі (пропускатися).

Проста умова містить два вирази (значення), поєднані знаком операції відношення:

$>$ більше за…
$<$ менше за…
$\geq$ більше або дорівнює…
$\leq$ менше або дорівнює …
$\neq$ не дорівнює…

Результатом перевірки умови є логічний вираз ІСТИНА, якщо умова виконується, або БРЕХНЯ, якщо умова не виконується.

Приклад 3.1. Знайти значення дійсних коренів квадратного рівняння

$$ax^2 + bx + c = 0 \ , \quad a \neq 0$$

Словесний спосіб подання алгоритму:

1. Уведення значень коефіцієнтів **a, b, c.**
2. Обчислення значення дискримінанта за формулою $\mathbf{D = b^2 - 4ac}$.
3. Перевірка отриманого значення дискримінанта: якщо дискримінант ≥ 0, то перехід на п.4, в противному разі перехід на п.6.
4. Обчислення дійсних коренів рівняння за формулами:

$$x_1 = \frac{-b + \sqrt{D}}{2a}$$

$$x_2 = \frac{-b - \sqrt{D}}{2a}.$$

5. Виведення отриманих результатів $\mathbf{x_1}$ та $\mathbf{x_2}$. Кінець алгоритму.
6. Виведення повідомлення «Дійсних коренів немає». Кінець алгоритму.

На рисунку 3.1. наведена блок-схема алгоритму:

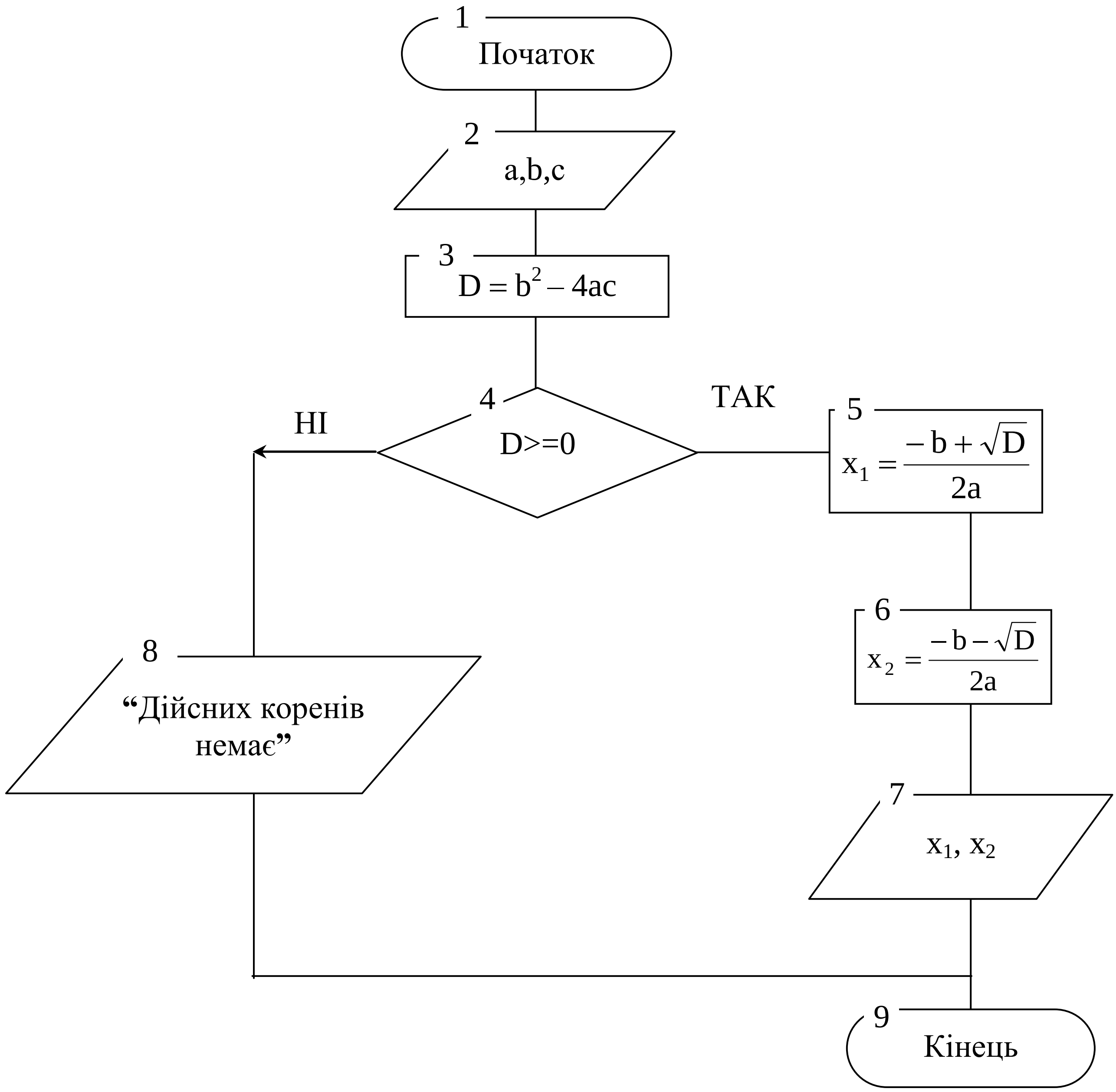


Рис. 3.1. Блок-схема алгоритму прикладу 3.1.

У блок-схемі алгоритму блок №4 використовується для перевірки умови і реалізує розгалуження: якщо умова набуває результату **істина**, то алгоритм продовжується по блоках 5, 6, 7, 9, а блок 8 зовсім не виконується. Навпаки, коли умова набуває значення **брехня**, алгоритм продовжується по блоках 8, 9, при цьому блоки 5, 6, 7 не виконуються.

Приклад 3.2. Для заданих значень ***x, a, b*** обчислити значення виразу:

$$Y = \begin{cases} \dfrac{abx^2}{b+x}, & x <= 2 \\ a + \sin(bx), & 2 < x <= 8 \\ \sqrt{|ab + x|}, & x > 8 \end{cases}$$

На рисунку 3.2. наведена блок-схема алгоритму.

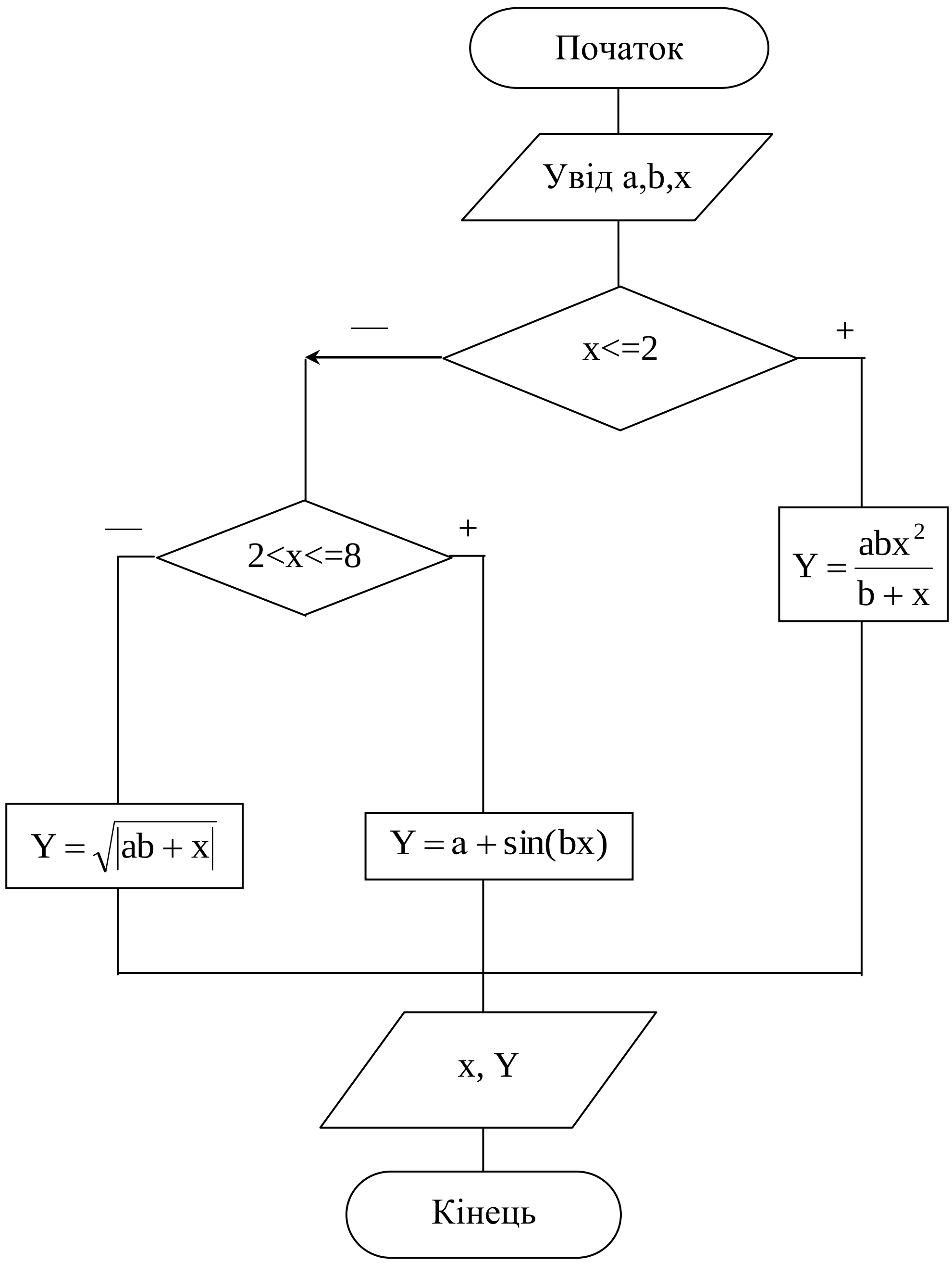


Рис. 3.2. Блок-схема алгоритму прикладу 3.2.

4. АЛГОРИТМИ ЦИКЛІЧНОЇ СТРУКТУРИ

Циклом називають частину алгоритму, яка повторюється.

При кожному черговому виконанні циклу перевіряється умова на продовження роботи і, якщо умова набуває результату ІСТИНА, цикл виконується, а якщо умова набуває результату БРЕХНЯ – цикл не виконується.

Перевірка умови може бути організована на початку циклу, і такий цикл називається циклом з передумовою, або у кінці циклу – тоді такий цикл називається циклом з післяумовою.

Різниця між такими циклами полягає в тому, що цикл з післяумовою виконується хоча б один раз, а цикл з передумовою може не виконуватися жодного разу.

Цикл по лічильнику характерний тим, що заздалегідь відома кількість повторень циклу, і цикл буде виконуватися, доки значення лічильника циклу не перевищить зазначену кількість повторень.

Якщо відомі початкове та кінцеве значення параметра циклу, а також закон (формула), за яким це значення змінюється, то цикл буде виконуватися, доки параметр циклу лежатиме у межах від початкового до кінцевого значення.

<u>Приклад 4.1.</u> Побудувати таблицю значень функції

$$Y = a + \sin(bx)$$

для заданих коефіцієнтів **a** і **b** та аргументу **x**, що змінюється від –4 до 6 з кроком 2.

<u>Словесний спосіб подання алгоритму</u>:

1. Уводяться коефіцієнти **a** і **b**.
2. Задається початкове значення аргументу $x = -4$.
3. Обчислюється значення функції **Y** для поточного аргументу.
4. Виводиться отримане значення функції **Y**.
5. Значення аргументу **x** збільшується на 2.
6. Перевіряється умова продовження циклу: якщо нове значення аргументу не перевищує заданого кінцевого значення **6**, то цикл (пункти – 6) виконується ще раз, у противному випадку — кінець алгоритму.

На рисунку 4.1. подана блок-схема алгоритму.

Цей цикл є циклом з післяумовою, тому що перевірка умови проводиться у кінці циклу.

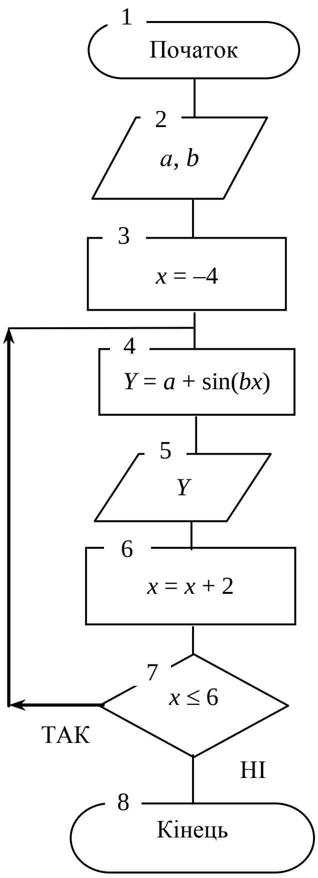


Рис. 4.1. Блок-схема прикладу 4.1.

Приклад 4.2. Обчислити значення функції

$$Y = \frac{b^2}{x^2 + b},$$

де x змінюється від $x_{\text{початкове}} = 0$ до $x_{\text{кінцеве}} = 3$ з кроком $\Delta x = 0{,}1$; $b = 3{,}8$

Число повторень циклу **n** обчислюється за формулою

$$n = \left[\frac{x_k - x_n}{\Delta\chi} \right] + 1$$

У нашому випадку цикл буде працювати 31 раз, тобто отримаємо 31 значення функції **Y**. Блок-схема алгоритму може бути наведена двома способами (рисунок 4.2. а, б)

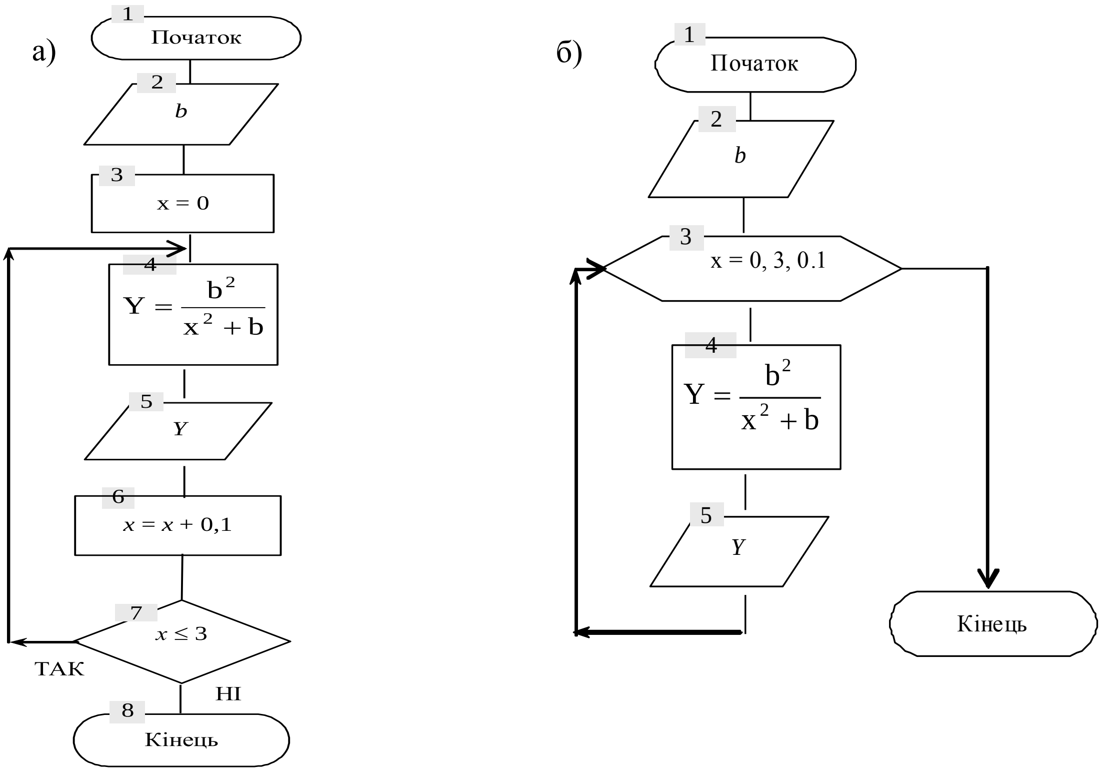


Рис. 4.2. Блок-схеми прикладу 4.2.

Приклад 4.3. Обчислити значення функції

$$Y = \begin{cases} \alpha \sin x, & x \le 0{,}2 \\ \cos x + 1, & x > 0{,}2 \end{cases}$$

де $x \in [-5;5]$, $\Delta x = 0.2$, $\alpha = 5x$

Блок-схема алгоритму наведена на рисунку 4.3.

5. ІТЕРАЦІЙНИЙ ЦИКЛ

Ітераційний цикл характерний тим, що виконується, доки проміжне значення не досягне зазначеної величини, тобто кількість повторень циклу заздалегідь невідома.

Приклад 5.1. Обчислити значення членів нескінченного ряду

$x;\ \frac{x^2}{2!};\ \frac{x^3}{3!};\ldots\frac{x^n}{n!},\ldots..$ де x — задане число

Обчислення проводити, доки не виконається умова $\frac{x^n}{n!} \le \varepsilon^{(1)}$, де ε – задане значення (точність обчислення).

У даному випадку ми не знаємо, при якому значенні **n** виконається умова (1).

Число повторень циклу залежить від проміжного результату, де:

$y_1 = x$ – перший член ряду;

$y_2 = y_1 \cdot \frac{x}{2}$ – другий член ряду;

$y_3 = y_1 \cdot \frac{x}{3}$ – третій член ряду;

……………

$y_n = y_{n-1} \cdot \frac{x}{n}$ – рекурентна формула для n-го члена ряду

Для того, щоб використати цю формулу для 1-го члена ряду $y_1 = y_0 \cdot \frac{x}{1}$ необхідно, щоб $y_0 = 1$.

На рисунку 5.1. подана блок-схема алгоритму.

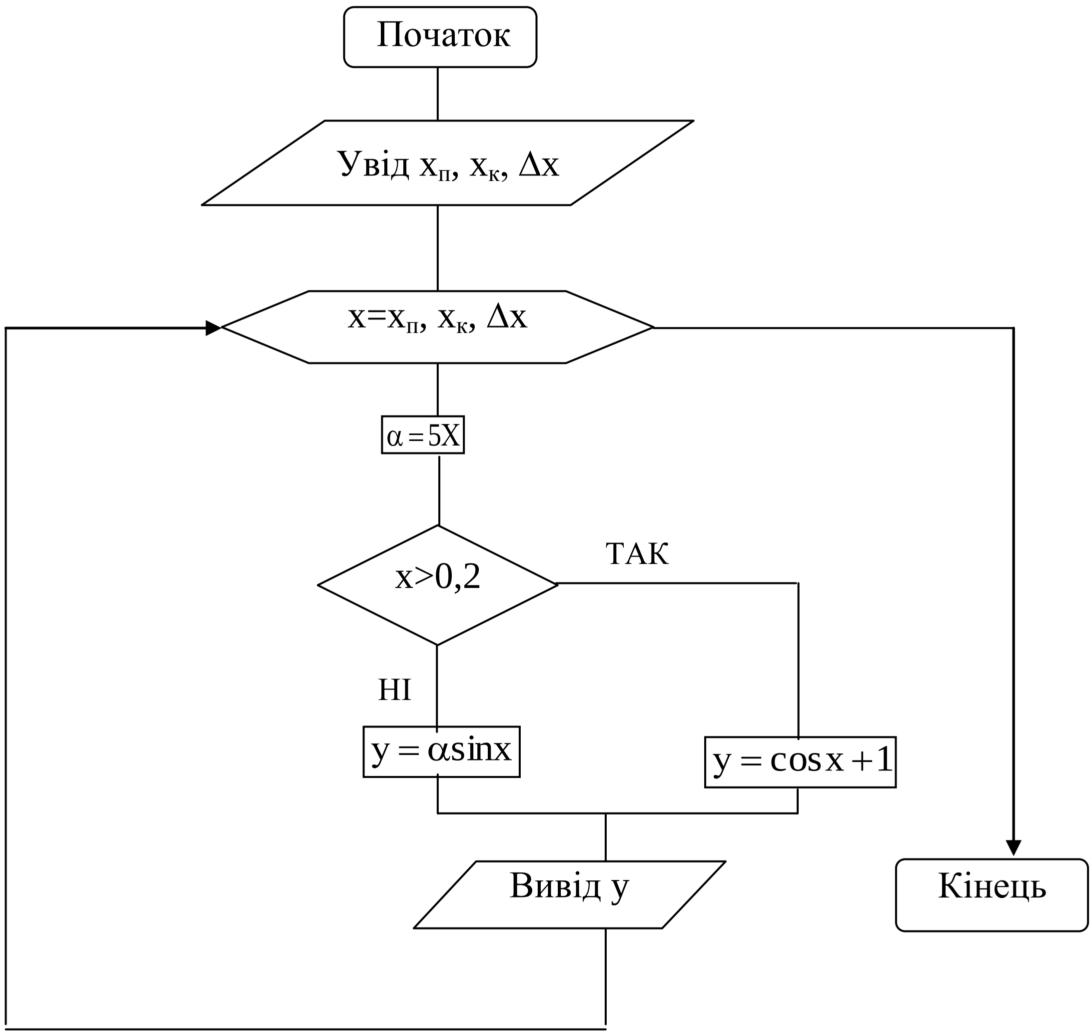


Рис. 4.3. Блок-схема прикладу 4.3.

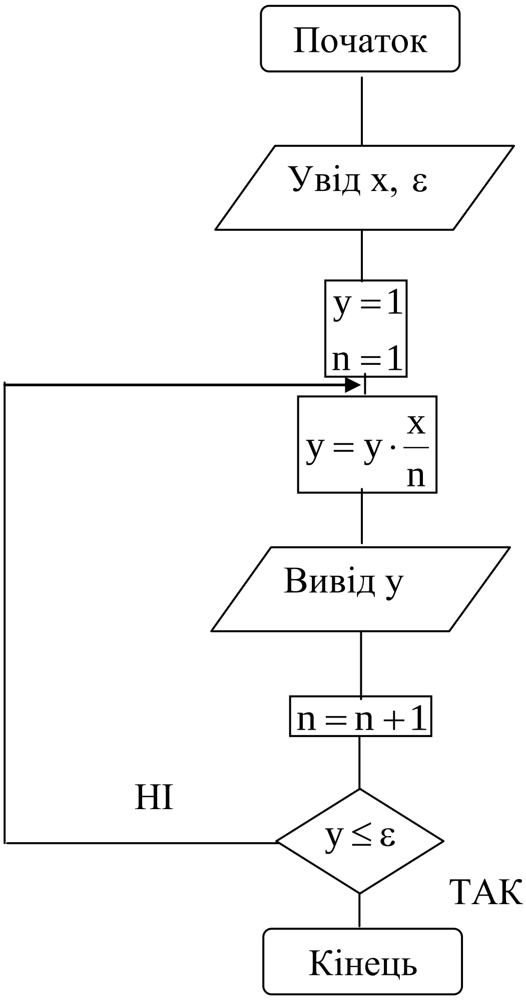


Рис. 5.1. Блок-схема прикладу 5.1.

Приклад 5.2. Обчислити вираз

$$Y = \sqrt[n]{100}, \quad \text{де} \quad n = 2, 3, \ldots$$

Визначити таке **n**, при якому $Y \le \varepsilon$,
якщо $\boldsymbol{\varepsilon}$ – задане значення (точність обчислення).

Словесний спосіб надання алгоритму:

1. Задається початкове значення **n** = 2.
2. Обчислюється функція **Y**.
3. Проводиться перевірка, чи отримане значення функції дорівнює або менше за $\boldsymbol{\varepsilon}$.
4. Якщо результат перевірки умови ІСТИНА, то виводиться поточне значення **n** та отримане значення функції, цикл припиняється. Кінець алгоритму.
5. Якщо результат перевірки умови БРЕХНЯ, значення **n** збільшується на одиницю (n + 1), і цикл повторюється з п.2.

На рисунку 5.2. подана блок-схема алгоритму.

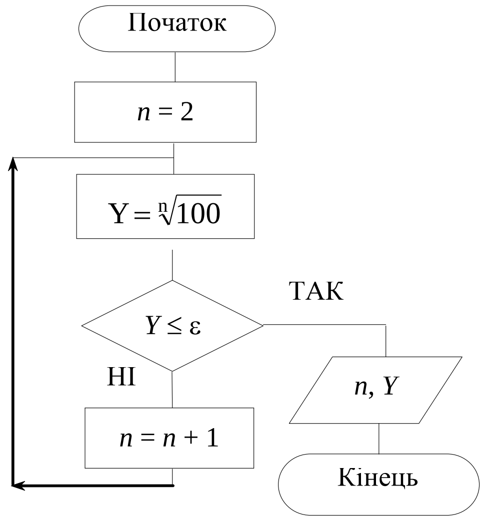


Рис. 5.2. Блок-схема прикладу 5.2

Приклад 5.3. Задано натуральне число N. Визначити кількість цифр у ньому. Блок-схема алгоритму наведена на рисунку 5.3.

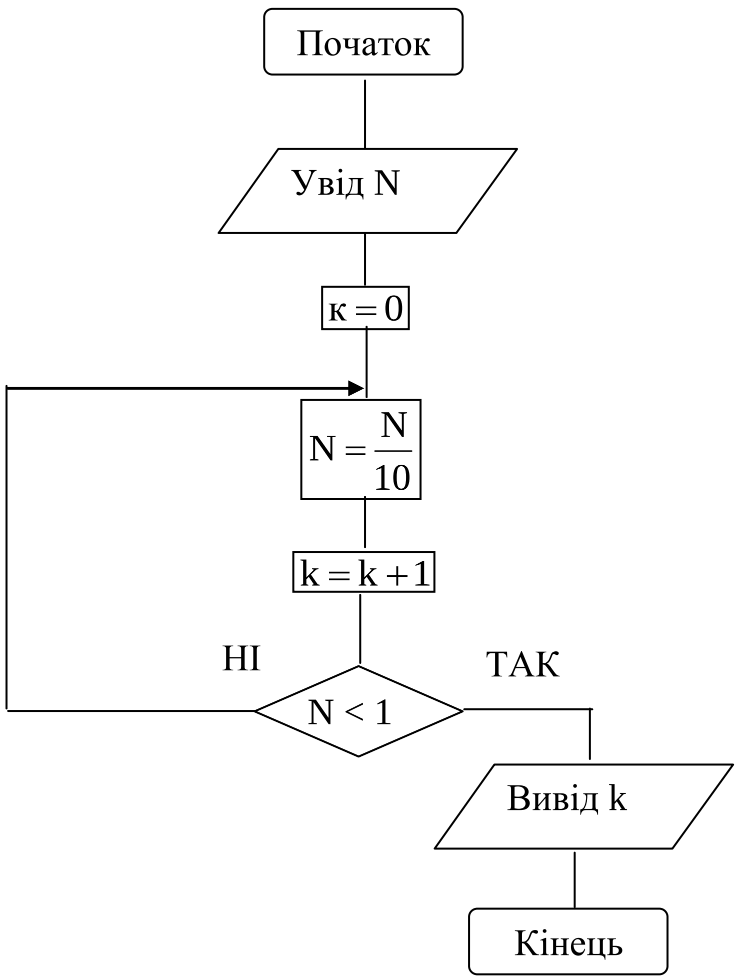


Рис. 5.3. Блок-схема прикладу 5.3.

Задача вирішується шляхом ділення числа на 10. Цикл буде працювати до тих пір, поки число не стане менше 1. К − кількість цифр у числі.

6. ОДНОМІРНІ МАСИВИ

6.1. Табулювання функції

Приклад 6.1. Побудувати таблицю значень функції

$$y = \frac{b^2}{x^2 + b},$$

де X – масив чисел, $X = (x_1, x_2, \dots x_n)$, $b = 4{,}8$.

На рисунку 6.1. зображено блок-схему алоритма.

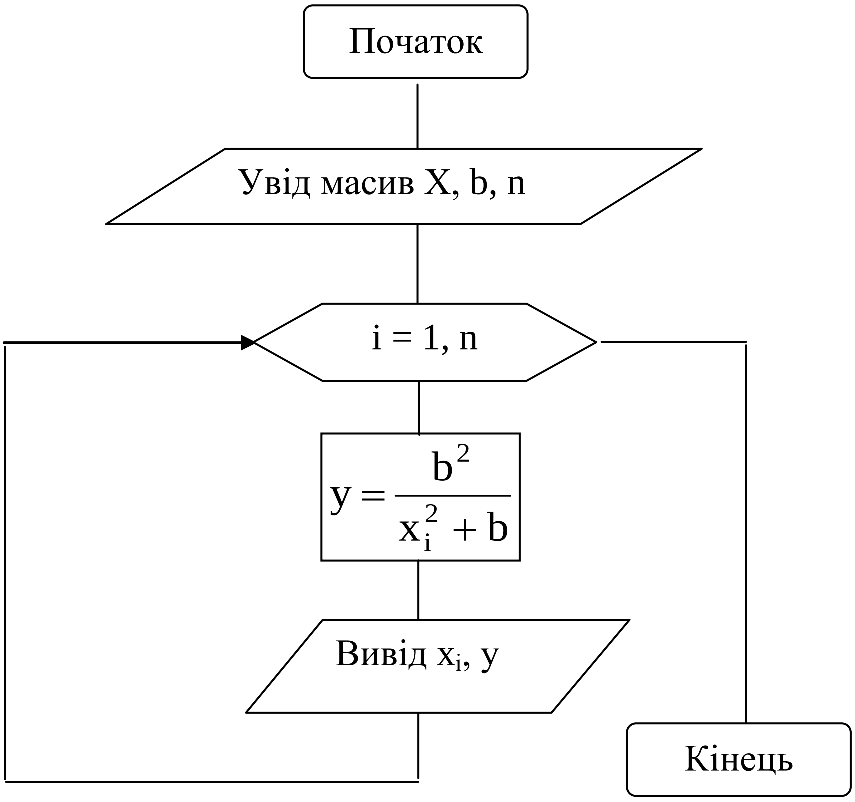


Рис. 6.1. Блок-схема прикладу 6.1

Нехай $X = (3; -0{,}8; 16{,}4; 100; -17; 2{,}4)$.
У даному випадку цикл буде працювати 6 разів, i – параметр циклу.

6.2. Накопичення суми і добутку елементів масиву

Приклад 6.2.1. Обчислити суму елементів масиву $\alpha = (\alpha_1, \alpha_1 \ldots \alpha_n)$.

$$S = \sum_{i=1}^{n} \alpha_i = \alpha_1 + \alpha_2 + \ldots + \alpha_n \qquad \text{(рис. 6.2.1.)}$$

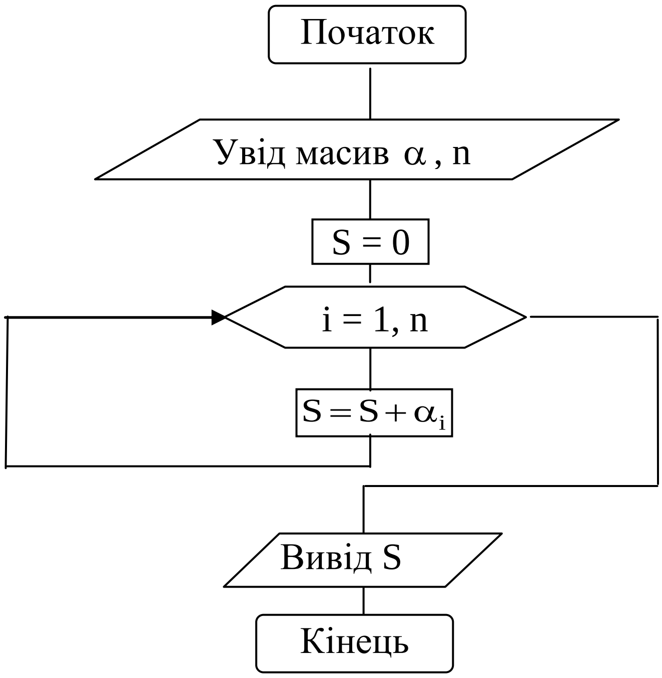


Рис. 6.2.1. Блок-схема алгоритма прикладу 6.2.1.

Приклад 6.2.2. Обчислити добуток елементів масиву $\alpha = (\alpha_1, \alpha_1 \ldots \alpha_n)$.

$$P = \prod_{i=1}^{n} \alpha_i = \alpha_1 \cdot \alpha_2 \cdot \ldots \cdot \alpha_n . \qquad \text{(Рис. 6.2.2.)}$$

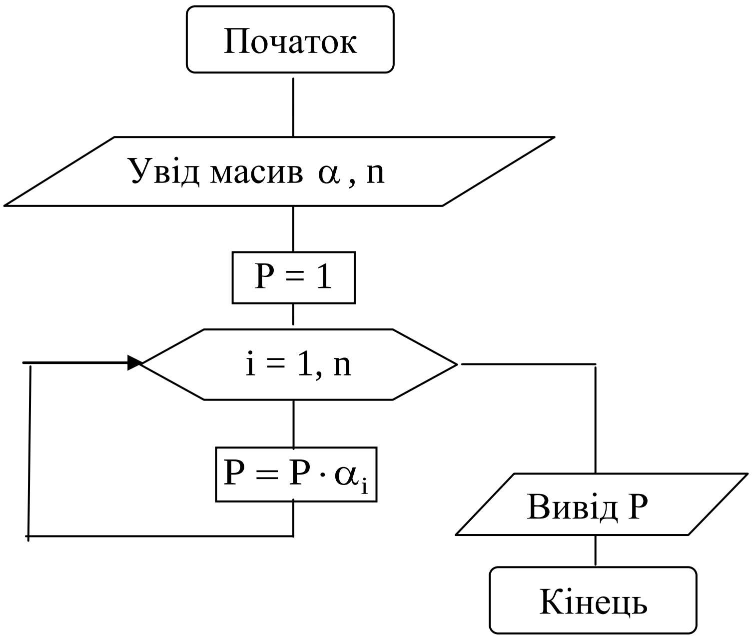


Рис. 6.2.2. Блок-схема алгоритма прикладу 6.2.2

6.3. Пошук максимального і мінімального елементів масиву

Приклад 6.3. Дан масив чисел $X = (x_1, x_1 \ldots x_n)$. Визначити максимальний елемент і зафіксувати його номер (рисунок 6.3).

Позначення: max – максимальний елемент масиву;

к – номер максимального елементу в масиві.

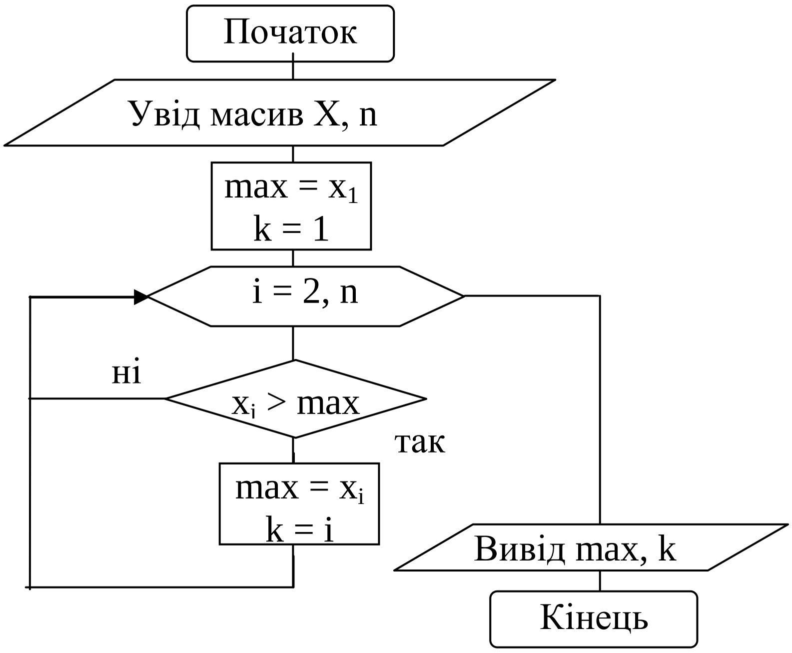


Рис. 6.3. Алгоритм пошуку максимального елементу та його номеру

Для визначення мінімального елементу масиву, слід внести зміни у блок-схему самостійно.

6.4. Обчислення середньоарифметичного та середньогеометричного елементів масиву

Приклад 6.4. Дан масив чисел $X = (x_1, x_1 \ldots x_n)$. Обчислити

а) середньоарифметичне значення додатних елементів;

в) середньогеометричне значення елементів більших за **1**.

На рисунку 6.4. зображені блок-схеми алгоритмів

Позначення: S – сума елементів масиву

P– добуток елементів масиву

K– кількість елементів масиву

SA– середньоарифметичне елементів масиву

SG – середньогеометричне елементів масиву

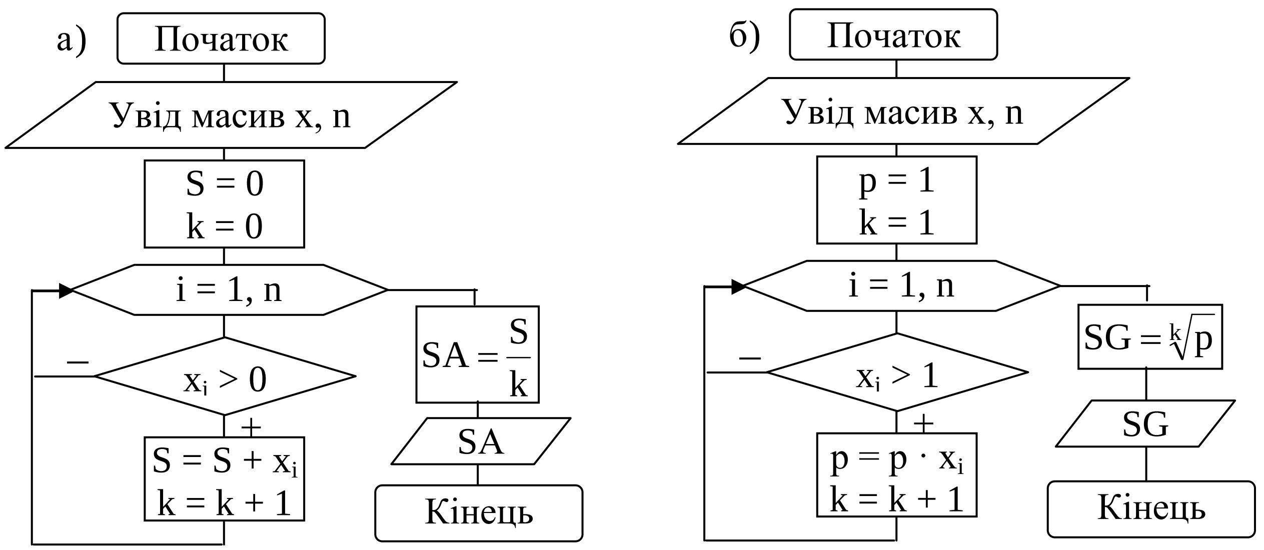


Рис. 6.4. а) Обчислення середньоарифметичного додатних елементів масиву;
б) Обчислення середньогеометричного елементів більших за **1**

6.5. Парні елементи, парні індекси елементів масиву

Приклад 6.5.1. Дан масив $X = (x_1, x_1 \ldots x_n)$. Збільшити усі парні елементи масива у 2 рази.

Скористуємось визначенням цілої частки числа. Ціла частка числа – це найбільше ціле, яке не більше даного числа.

Наприклад: $[3{,}5] = 3$, $[3{,}9] = 3$, $[-2{,}2] = -3$.

[ ] – визначення цілої частки числа.

На рисунку 6.5.1. зображена блок-схема алоритма

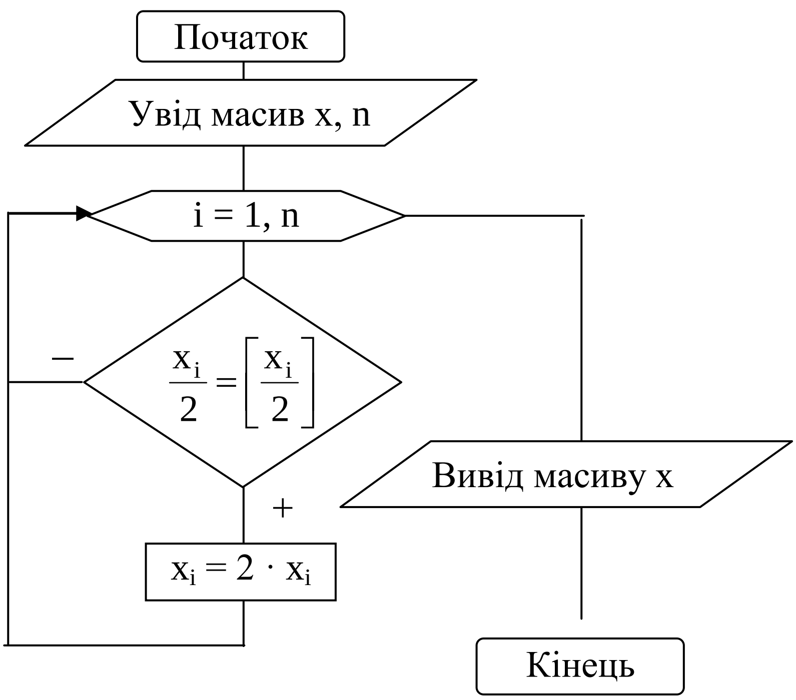


Рис. 6.5.1. Блок-схема алгоритму прикладу 6.5.1.

Приклад 6.5.2. Дан масив $X = (x_1, x_1 \ldots x_n)$. Елементи з парними номерами замінити на (-1), а останні поділити на 1-ий елемент масиву.

На рисунку 6.5.2. зображена блок-схема алоритма.

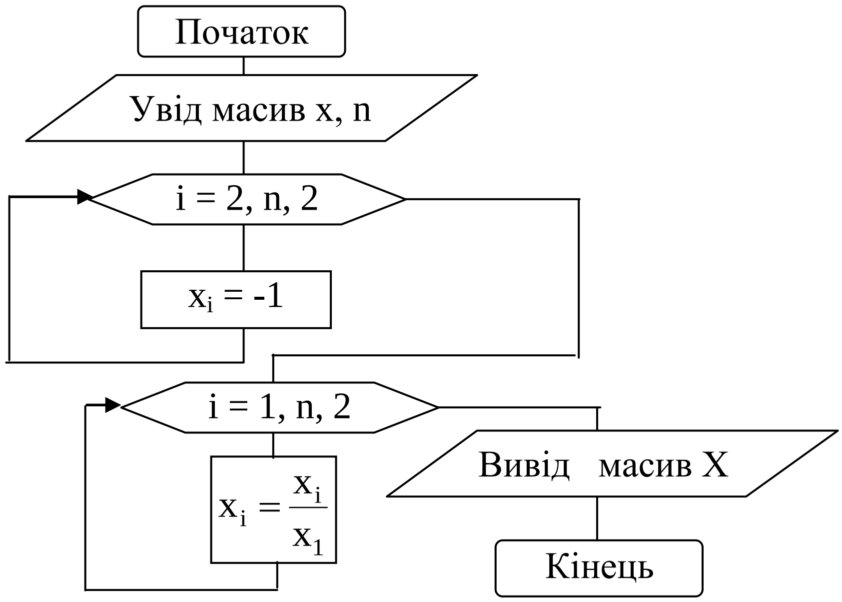


Рис. 6.5.2. Блок-схема алгоритма прикладу 6.5.2.

6.6. Сортування елементів масиву

Приклад 6.6. Дан масив $A = (a_1, a_2, \ldots a_n)$.
Отсортувати значення його елементів по зростанню.

Будемо зрівнювати два поруч розташованих елемента, якщо наступний елемент менший попереднього, то переставимо їх місцями.

Пояснення:

$p = 0$ – признак того, що перестановок не було;

$p = 1$ – признак того, що була хоча б одна перестановка.

На рисунку 6.6. зображена блок-схема алгоритму.

Алгоритм сортування елементів масиву по зменшенню їх значень скласти самостійно.

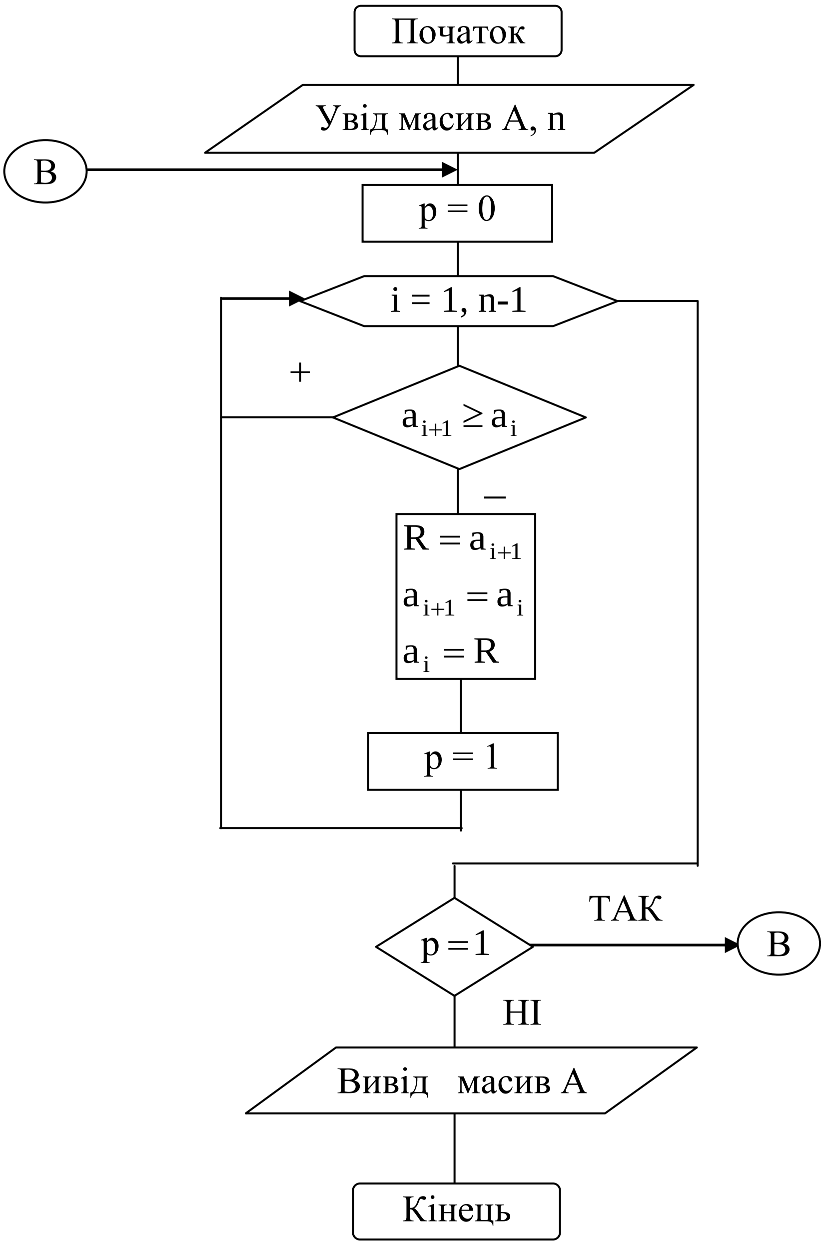


Рис. 6.6. Блок-схема алгоритму сортування елементів масиву по зростанню

6.7. Перезапис елементів одного масиву в іншій за заданою умовою

Приклад 6.7.1. Дан масив $A = (a_1, a_2, \ldots a_n)$.
Переписати у масив **B** ті елементи масиву **A**, які мають парні індекси.

Пояснення:

i – номер елементу в масиві B;
j – номер елементу в масиві A;
к – кількість елементів в масиві B.

На рисунку 6.7.1. зображена блок-схема алоритма.

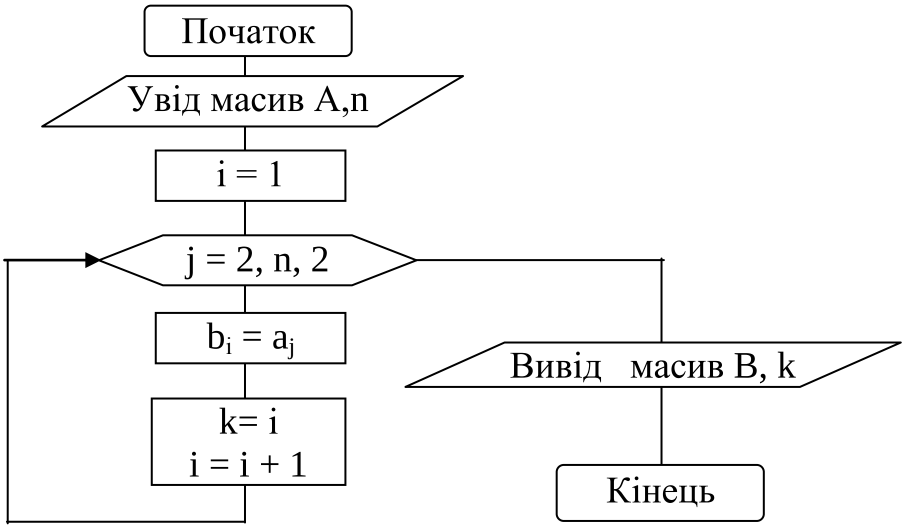


Рис. 6.7.1. Блок-схема алгоритму прикладу 6.7.1.

<u>Приклад 6.7.2.</u> Дан масив $A = (a_1, a_2, \ldots a_n)$. Переписати у масив **B** усі від'ємні елементи масиву **A.**

Пояснення:

i – номер елементу в масиві B;
j – номер елементу в масиві A;
к – кількість елементів в масиві B.

На рисунку 6.7.2. зображена блок-схема алоритма.

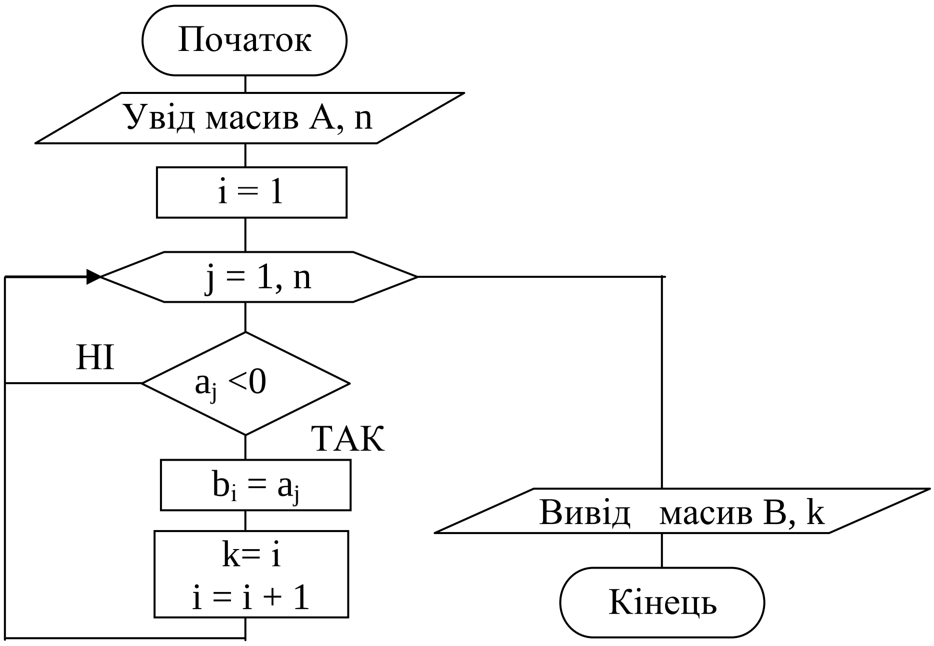


Рис. 6.7.2. Блок-схема алгоритму прикладу 6.7.2.

6.8. Приклади деяких алгоритмів

Приклад 6.8.1. Визначити найбільший з від’ємних елементів масиву $Y=(y_1, y_2,\dots y_{30})$.

На рисунку 6.8.1. зображена блок-схема алгоритму.

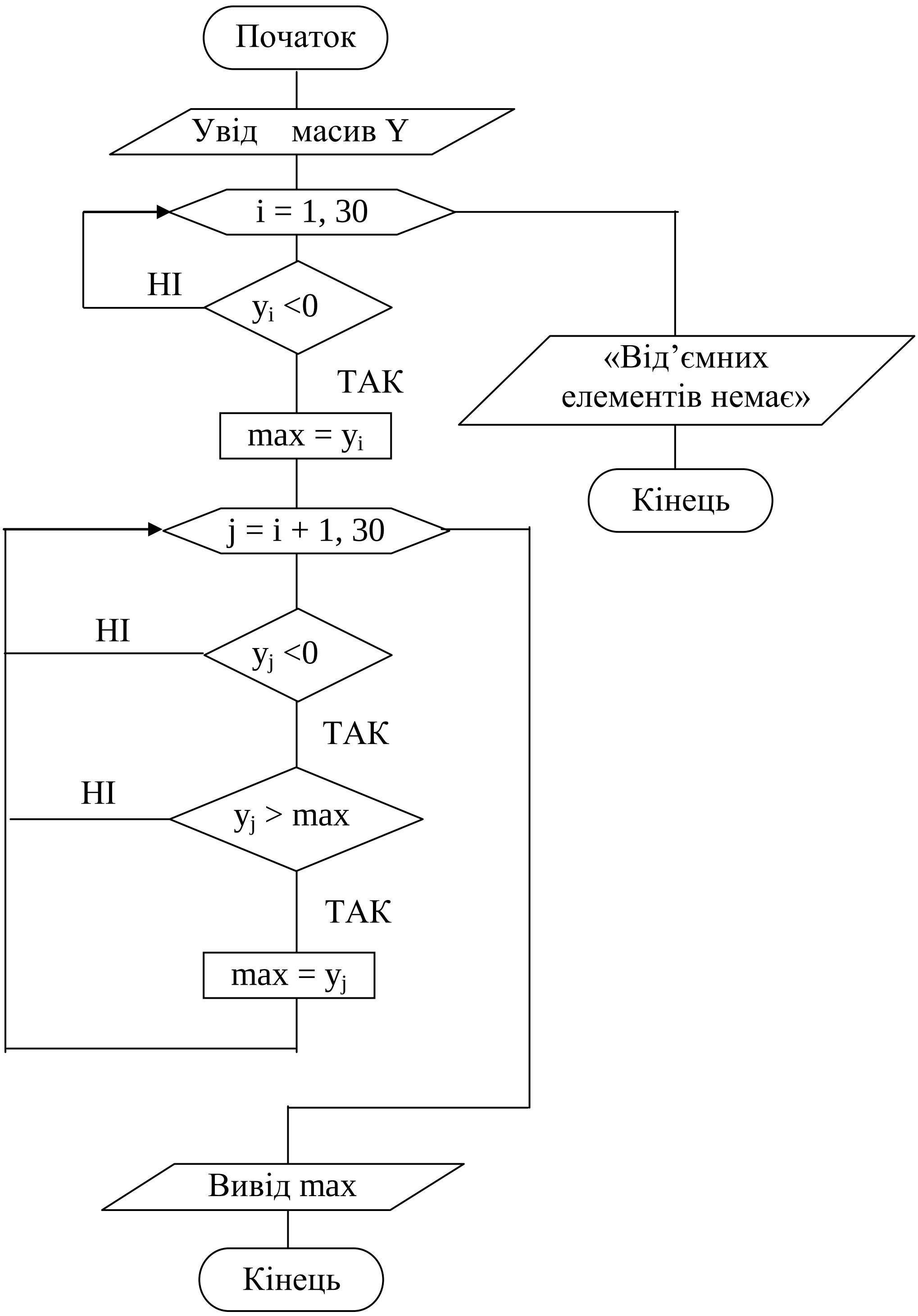


Рис. 6.8.1. Блок-схема алгоритму прикладу 6.8.1.

Приклад 6.8.2. Послідовність елементів $a_1, a_2, \ldots a_n$ задана формулою

$$a_i = 2^i + 3^{i+1} + i, \quad i = \overline{1,n}, \qquad n \text{ – задане число.}$$

Визначити кількість елементів кратних 3.

На рисунку 6.8.2. зображена блок-схема алгоритму.

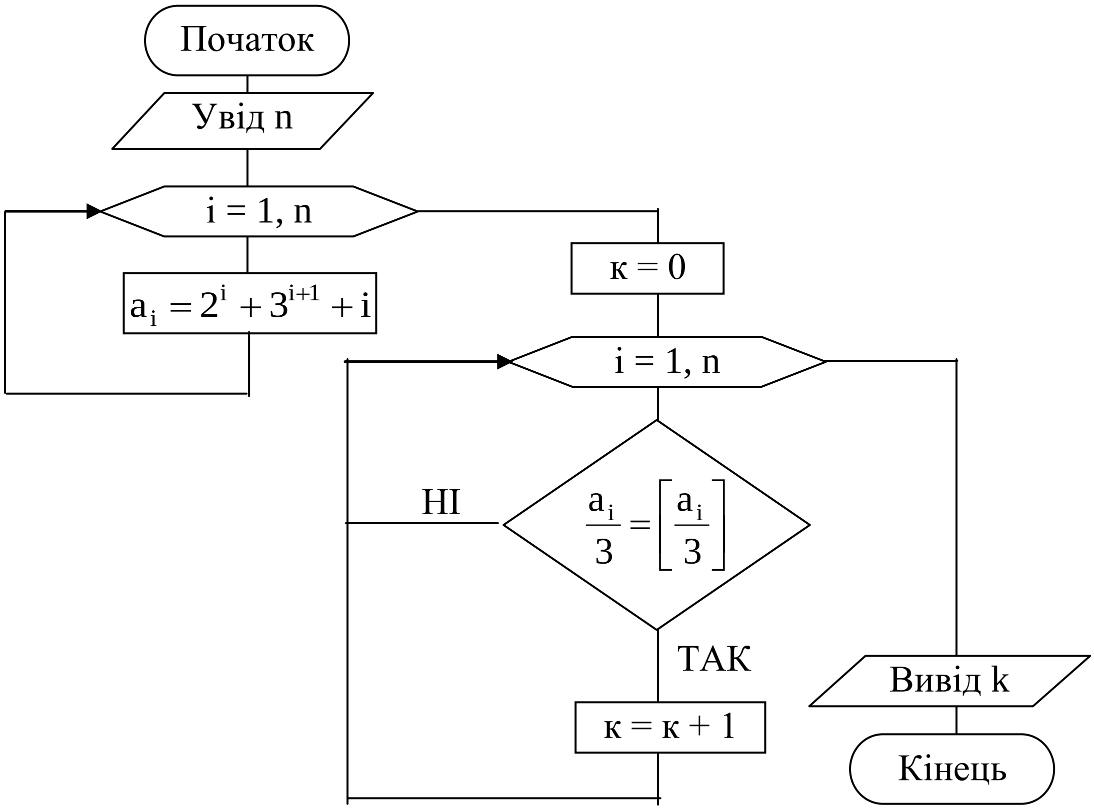


Рис. 6.8.2. Блок-схема алгоритму 6.8.2.

7. ОБЧИСЛЕННЯ СУМИ (ДОБУТКУ) РЯДУ

Приклад 7.1. Знайти суму ряду

$$\frac{y}{2} + \frac{y^2}{4} + \frac{y^3}{6} + \ldots + \frac{y^{10}}{20}, \quad \text{де y – задане число.}$$

На рисунку 7.1. зображена блок-схема алгоритму.

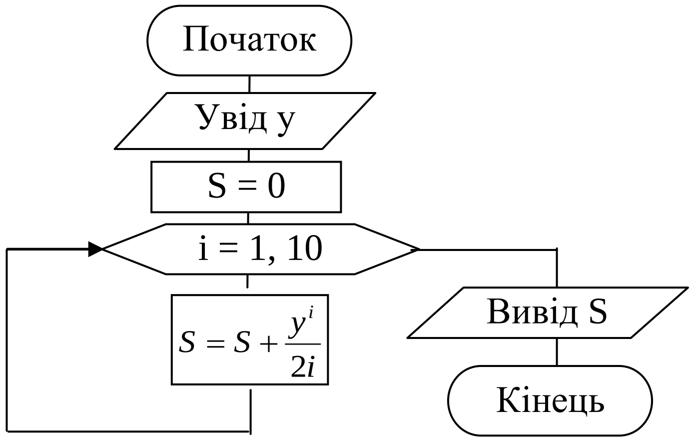


Рис. 7.1. Блок-схема алгоритму прикладу 7.1.

Приклад 7.2. Обчислити суму ряду

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\ldots+\frac{1}{999\cdot 1000}$$

На рисунку 7.2. зображена блок-схема алгоритму.

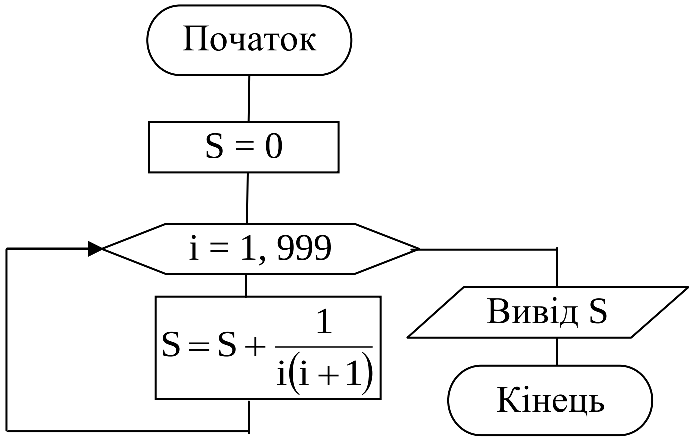


Рис. 7.2. Блок-схема алгоритму прикладу 7.2.

Приклад 7.3. Обчислити добуток за формулою

$$a(a-n)(a-2n)\ldots(a-n^2),$$

якщо a, n – задані числа.

На рисунку 7.3. зображена блок-схема алгоритму.

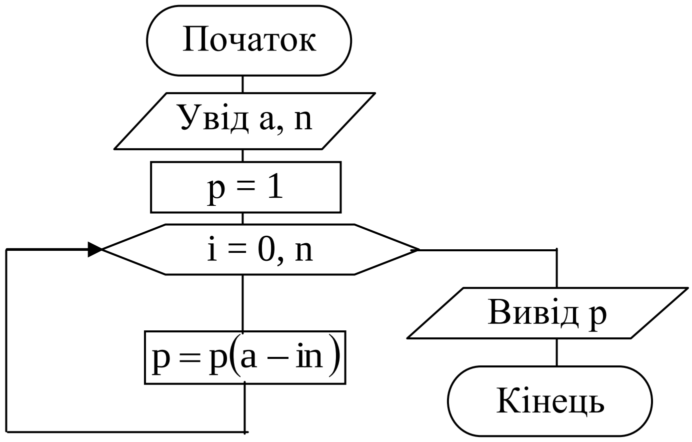


Рисунок 7.3. Блок-схема алгоритму прикладу 7.3.

8. ДВОМІРНІ МАСИВИ

Двомірний масив має вигляд:

$$A(m,n)=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad , \quad \text{де}$$

m – кількість рядків
n – кількість стовпців
a_{ij} – елемент масиву
i – номер рядка
j – номер стовпця

Приклад 8.1. Обчислити суму елементів у кожному рядку заданого масиву A(m, n).

m, n – задані значення.

На рисунку 8.1. зображена блок-схема алгоритму.

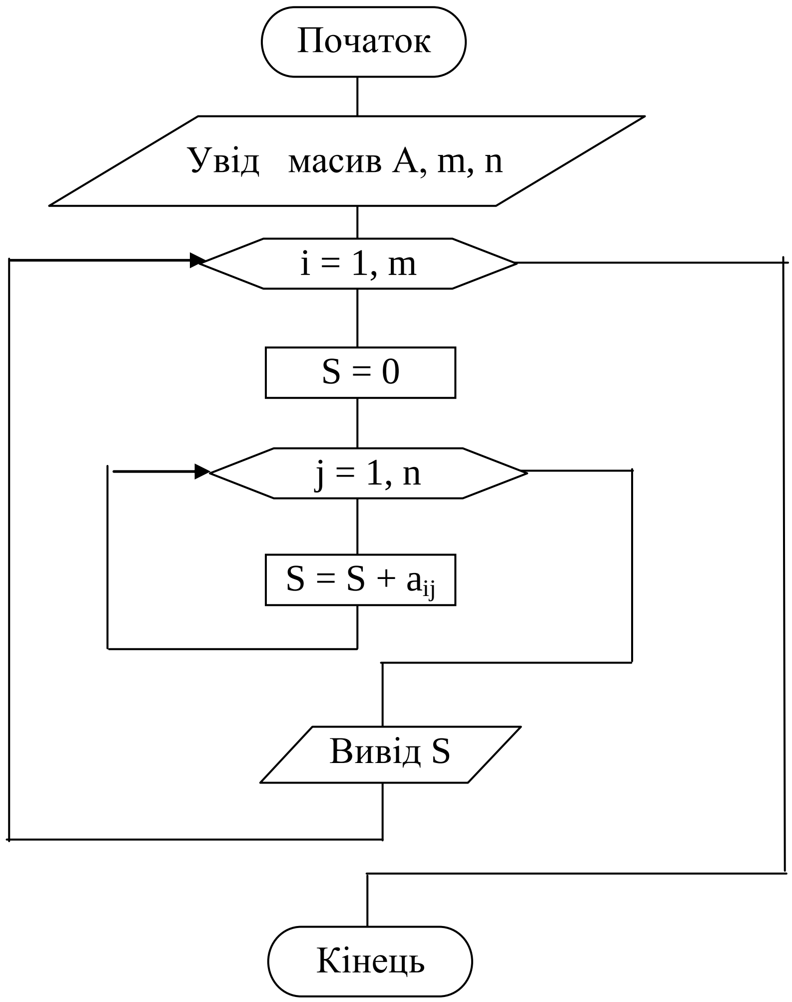


Рис. 8.1. Блок-схема прикладу 8.1.

Приклад 8.2. Визначити максимальний (max) елемент у кожному стовпцю двомірного масиву A(m,n).

На рисунку 8.2. зображена блок-схема алгоритму.

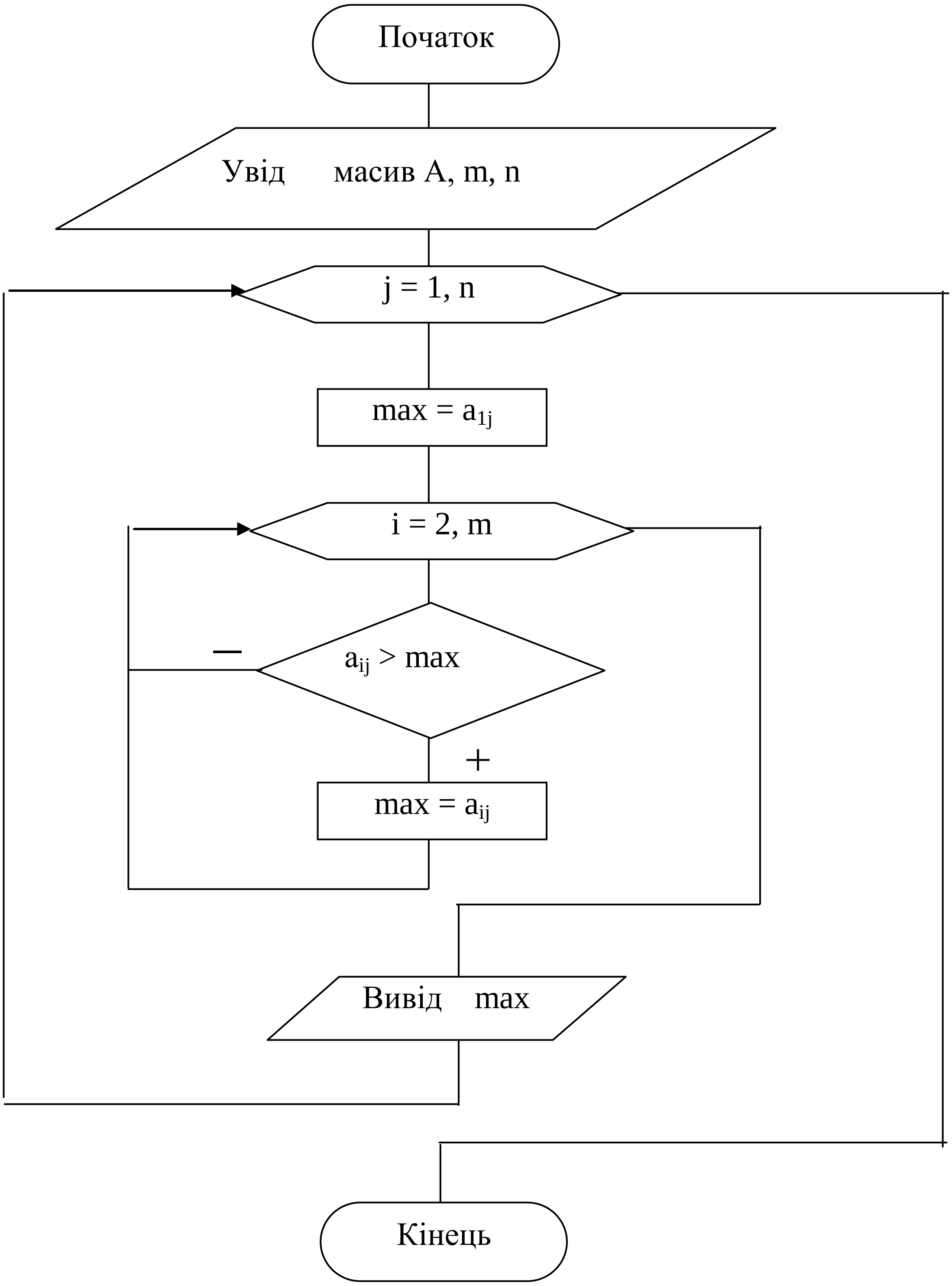


Рис. 8.2. Блок-схема алгоритму прикладу 8.2.

Приклад 8.3. Визначити суму елементів головної діагоналі і суму елементів додаткової діагоналі у квадратній матриці $B_{n,n}$.

Позначимо: S_1 – сума елементів головної діагоналі матриці
S_2 – сума елементів додаткової діагоналі матриці

На рисунку 8.3. зображена блок-схема алгоритму.

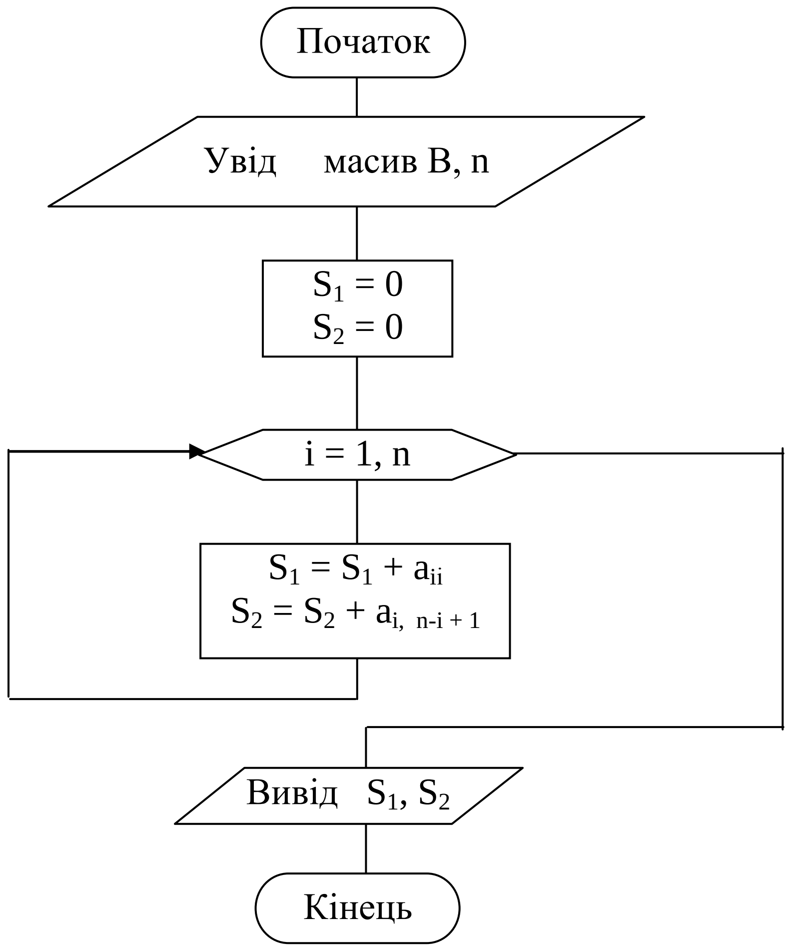


Рис. 8.3. Блок-схема алгоритму прикладу 8.3.

9. ПІДПРОГРАМИ

При створенні додатків часто застосовується структурне програмування. При цьому програма розбивається на підпрограми (підалгоритми), кожна з яких виконує одну з дій, які передбачені умовами задачі.

Важлива характеристика підпрограм — це можливість їх повторного виконання.

Підпрограми бувають 2-х видів — процедури та функції.

9.1. Підпрограми процедури

Приклад 9.1. Існують масиви $X = (x_1, x_2, \ldots x_{10})$ та $Y = (y_1, y_2, \ldots y_{20})$. Знайти суми додатних елементів у кожному з них.

На рисунку 9.1. зображені блок-схеми алгоритму.

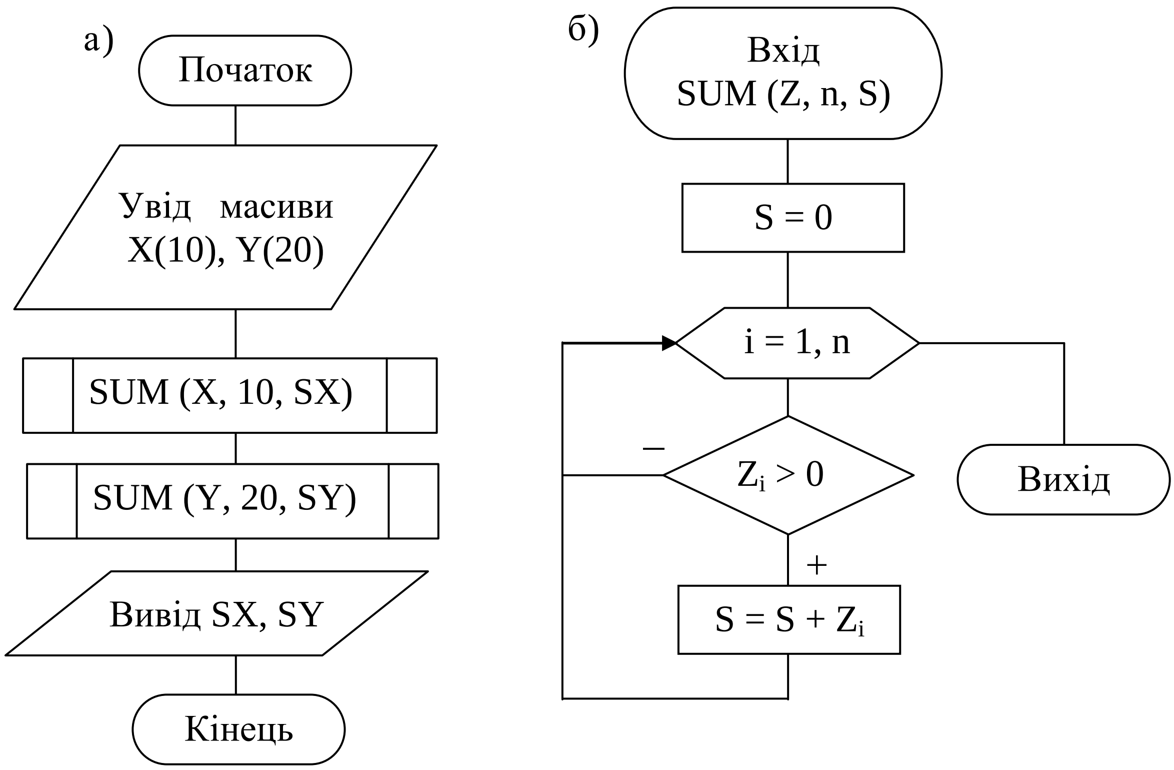


Рис. 9.1. Блок-схеми алгоритму прикладу 9.1.
а) блок-схема до головної програми
б) блок-схема до підпрограми процедури

SUM – ім’я підпрограми процедури
X, 10, SX – фактичні параметри
SX – сума додатних елементів масиву X
Y, 20, SY – фактичні параметри
SY – сума додатних елементів масиву Y

Z, n, S – <u>формальні параметри</u>

Z – одномірний масив

n – кількість елементів у масиві Z

S – сума додатних елементів у масиві Z

<u>Приклад 9.2.</u> Знайти кількість від’ємних елементів у кожному стовбцю заданого двомірного масиву A(3,4).

На рисунку 9.2. зображени блок-схеми алгоритму.

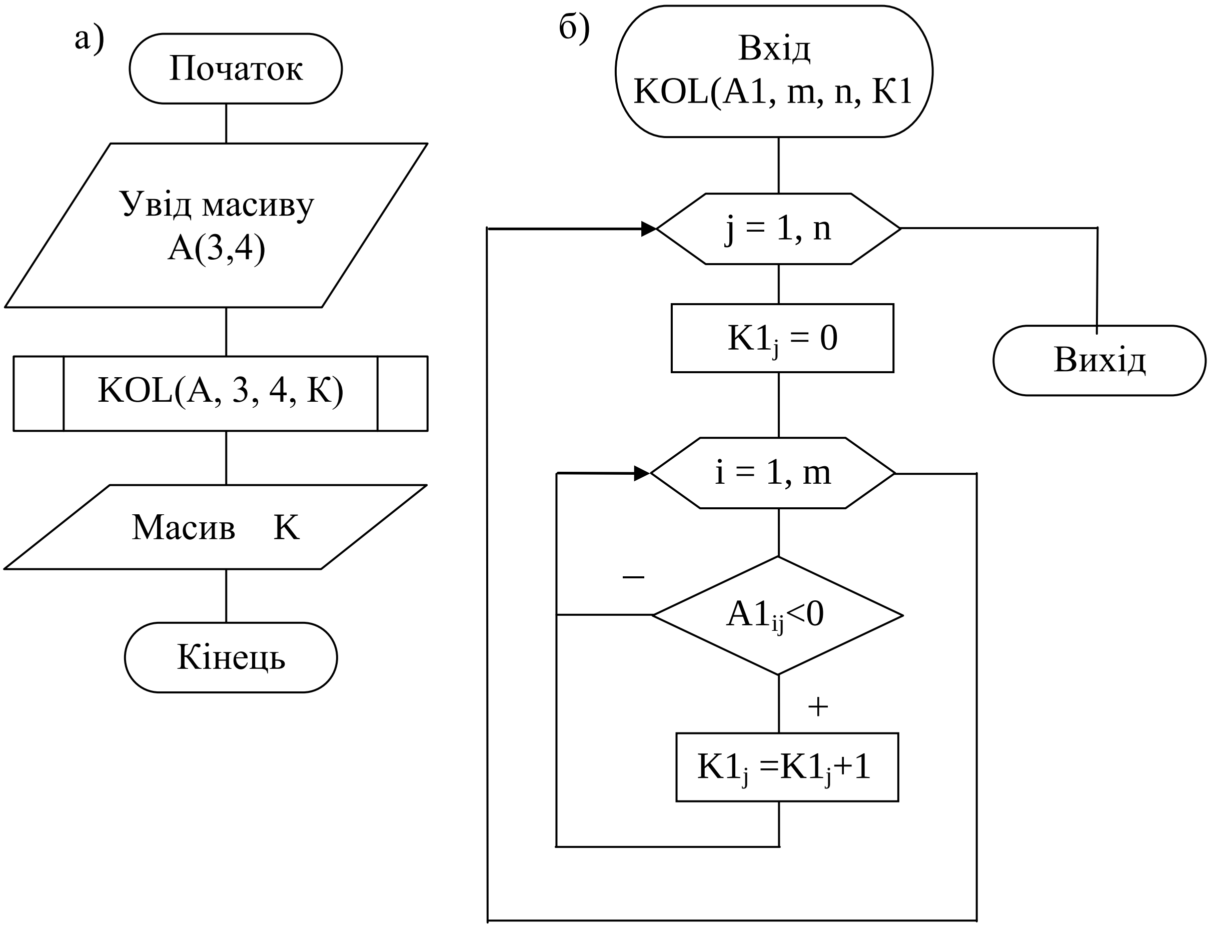


Рис. 9.2. Блок-схеми алгоритму прикладу 9.2.

а) блок-схема до головної програми

б) блок-схема до підпрограми процедури

KOL – ім’я підпрограми процедури

A, 3, 4, K – <u>фактичні параметри</u>

K – ім’я одномірного масиву, який утримує кількість від’ємних елементів в кожному стовбці заданого масиву A(3,4).

A1, m, n, K1 – <u>формальні параметри</u>

A1 – двомірний масив

m – кількість рядків

n – кількість стовбців

K1 – одномірний масив

9.2. Підпрограми функції

Приклад 9.3. Існують масиви $X=(x_1, x_2, \ldots x_{10})$ та $Y=(y_1, y_{2,} \ldots y_{20})$. Знайти максимальний елемент в кожному з них.

На рисунку 9.3. зображени блок-схеми алгоритму.

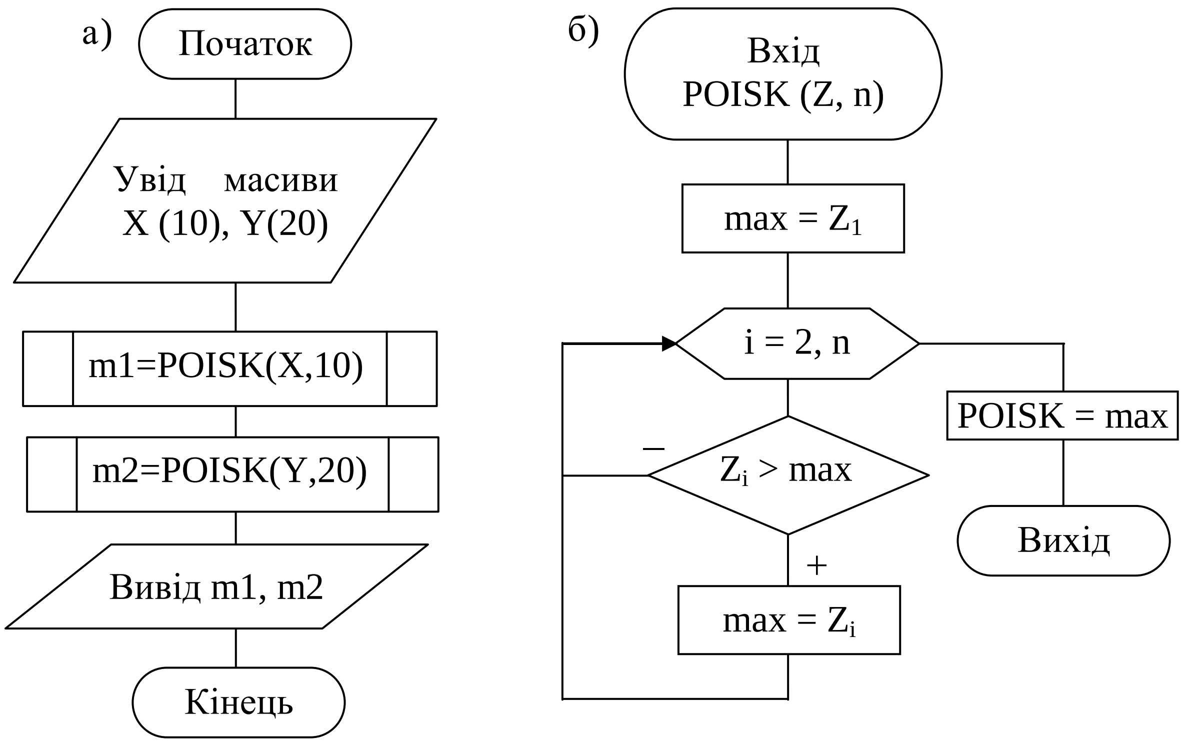


Рис. 9.3. Блок-схеми алгоритму прикладу 9.3.
а) блок-схема до головної програми
б) блок-схема до підпрограми функції

POISK – ім’я функції
X, 10 – фактичні параметри
m1 – максимальний елемент у масиві X
Y, 20 – фактичні параметри
m2 – максимальний елемент у масиві Y
Z, n – формальні параметри
Z – одномірний масив
n – кількість елементів у масиві Z

Приклад 9.4. Створити підпрограму функцію для обчислення значення функції:

$$Z=\begin{cases}1+3{,}8\cos x, & x\geq 2\\ \sqrt{x^2+2y^2}, & x<2\end{cases}$$

$$x=b\sin\alpha;\quad y=8{,}6+x^2;\quad b=13{,}4;\quad \alpha\in[0{,}1;55];\quad \Delta\alpha=1{,}2.$$

На рисунку 9.4. зображени блок-схеми алгоритму.

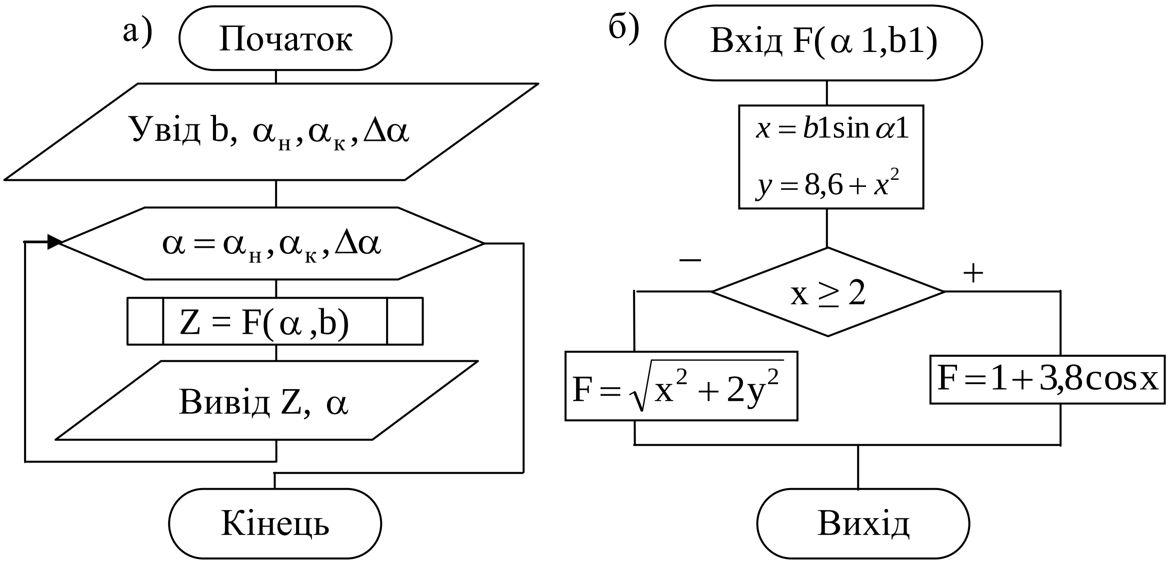


Рис. 9.4. Блок-схеми алгоритму прикладу 9.4.
а) блок-схема до головної програми
б) блок-схема до підпрограми функції

F – ім’я функції
α, b – фактичні параметри
α1, b1 – формальні параметри

10. ВАРІАНТИ ЗАВДАНЬ ДЛЯ САМОСТІЙНОЇ РОБОТИ

10.1. Алгоритми лінійної структури

Скласти блок-схеми алгоритмів обчислення значення функції Y за варіантами:

1. $y = \dfrac{x^4 - bx^3 - a}{(x+a)(x-b)},\quad \partial e\ b = 2^x;\ x = \lg 0{,}05 + 2;\ a = b + 2$

2. $y = \dfrac{x + 3a - k_1}{k_1 x + k_2},\quad \partial e\quad x = 7a + k_1 \cdot k_2;\ k_1 = 0{,}8;\ k_2 = 4{,}8;\ a = 0{,}25$

3. $y = \dfrac{\sin^3 ax + b}{\cos^2 x},\quad \partial e\quad x = -3{,}8;\ a = 0{,}5c + x^2;\ c = \ln 0{,}08;\ b = x^2 + c$

4. $y = \sin\dfrac{a}{x} + \lg 0{,}08e^x,\quad \partial e\quad x = -2{,}5a^2;\ a = 0{,}8b + c;\ c = 0{,}5;\ b = c^3$

5. $y = \dfrac{\left(a\sqrt[3]{b} - cd\right)^2}{a + b + c},\quad \partial e\ a = 7{,}14;\ b = a^2 - 1;\ c = a^3 - b^3;\ d = \sqrt{(b-c)^3}$

6. $y = \dfrac{\alpha_1\beta_1 - \alpha_2\beta_2}{\alpha_1 + \beta_1^2},\quad \partial e\ \alpha_1 = \sin 0{,}18;\ \beta_1 = e^4;\ \alpha_2 = \ln 0{,}5 + \beta_1;\ \beta_2 = 1{,}7 \cdot \beta_1$

7. $y = \dfrac{(a+b)^2 + m}{1 + x^a + b^a},\quad \partial e\ x = 0{,}81;\ a = \lg 0{,}83;\ b = e^a;\ m = x - 1$

8. $y = (1+z)\dfrac{x+y}{a-x},\quad \partial e\ x = 0{,}8 \cdot 10^{-2};\ y = e^{\sqrt{x^3}};\ z = 5 \cdot 10^{-3};\quad a = z + \dfrac{y}{2}$

9. $y = \dfrac{\sqrt{x^4 + ax + b}}{\sqrt[3]{x^4 - ax - b}},\quad \partial e\ x = 0{,}75;\ a = -x^2 + \lg 0{,}08;\quad b = e^{-x} + a$

10. $y = \ln\left(e^{x} + bx^{2}\right),\quad \partial e\ \ x = \sin^{2}(a+b);\ \ a = 0{,}36;\ \ b = \sin^{2} a$

11. $y = 5\sin^{2}\left|\ln(cx+1)\right|,\quad \partial e\ \ c = 4{,}8;\ a = 3{,}6;\quad x = \cos^{2}(a+b);\quad b = \sin a$

12. $y = \sqrt{x^{3} + ax^{2}} + c,\quad \partial e\ \ x = 0{,}36;\ \ a = x + 0{,}52b;\ \ b = ax^{2} + 1;\ \ c = 0{,}7$

13. $y = \cos\dfrac{1}{x + 0{,}2} + \lg 0{,}8x,\quad \partial e\ \ x = -2{,}6a^{2} + b\sqrt[3]{c};\ \ c = 7{,}14;\ \ a = 0{,}8c;\ \ b = ac^{2}$

14. $y = \dfrac{\left(b\sqrt[4]{c} - bd\right)^{2}}{a - b},\quad \partial e\ \ c = 0{,}13;\ \ b = c^{3} - 4a;\ \ a = 7{,}6;\ \ d = c^{2} + b^{2}$

15. $y = \sin^{3}\left(\dfrac{x + a}{2}\right) - \cos x,\quad \partial e\ \ x = 5b + e^{a};\ \ a = 4{,}8;\ \ b = \ln a$

10.2. Алгоритми розгалуженої структури

Скласти блок-схеми алгоритмів обчислення значення функції Y за варіантами:

1. $$y=\begin{cases} x^2-\sin\gamma, & x\le 0\\ \sqrt{x}+\cos\gamma, & x>0 \end{cases};$$
$\gamma=0{,}35a;\ a=x+3;\ x=1{,}8.$

2. $$y=\begin{cases} 4x^3+\gamma, & x\le 2\\ (x+3\gamma)\cdot x, & x>2 \end{cases}$$
$x=1{,}4;\ a=3{,}5x;\ \gamma=\cos(a+x).$

3. $$y=\begin{cases} x^3+a, & x<0\\ \sin\dfrac{x}{a}, & x=0\\ \sqrt{x}+\dfrac{a}{2}, & x>0 \end{cases}$$
$x=1{,}3-\ln a;\ a=0{,}27.$

4. $$y=\begin{cases} e^{-x+2}+a, & x\ge 1\\ \dfrac{\sin(x+3{,}2)}{a+3}, & x<1 \end{cases}$$
$a=14{,}8;\ x=e^a.$

5. $$y=\begin{cases} \mathrm{tg}\dfrac{1}{x^2+8{,}2}, & x\ge 0\\ \dfrac{0{,}32x}{x^2+3}, & x<0 \end{cases}$$
$x=0{,}5k+b;\ b=\sin 1{,}7;\ k=-0{,}8.$

6. $$y=\begin{cases} \dfrac{9bx}{x-2bx^2}, & x<2\\ \cos(b+x), & x\ge 2 \end{cases}$$
$x=\ln 0{,}7;\ b=0{,}3.$

7. $$y=\begin{cases} \cos x\text{ - }\sin^3 x, & x\ge 1{,}5\\ xe^{-x}, & x<1{,}5 \end{cases}$$
$x=0{,}9\cdot\sin a;\ a=e^{0{,}17}.$

8. $$y=\begin{cases} x(A-C), & A=C\\ x^3-A, & A>C\\ x^3+A, & A<C \end{cases}$$
$x=8{,}6;\ a=2\cos x;\ C=A+1.$

9. $$y=\begin{cases} 0{,}5\cos x, & x<1\\ 0{,}25x^a, & x=1\\ 0{,}9\sqrt{x}, & x>1 \end{cases}$$
$x=1{,}7-e^{0{,}35}.$

10. $$y=\begin{cases} 6z^2-5, & x\le 1\\ 5z^3+1, & x>1 \end{cases}$$
$z=0{,}5;\ x=\arccos z.$

11. $$y = \begin{cases} (4 - x)^2, & b < 5 \\ 0{,}25 + bx, & b \ge 5 \end{cases}$$

$x = 0{,}7;\ b = cx^2;\ c = \sin^2 x.$

12. $$y = \begin{cases} a + b - x, & x \ge 1{,}5 \\ \sin^2 x + \cos(a + b), & x < 1{,}5 \end{cases}$$

$a = 1{,}3;\ b = 0{,}85;\ x = 4{,}7 - \lg a.$

13. $$y = \begin{cases} 2{,}35 + x^3, & x \le 1{,}5 \\ \sqrt{0{,}85x^2}, & x > 1{,}5 \end{cases}$$

$x = 5a + \sin 3{,}8;\ a = 14{,}6.$

14. $$y = \begin{cases} \frac{a}{2}(1 - \alpha) + x^3, & \alpha \le 1 \\ 0{,}1\alpha^4, & \alpha > 1 \end{cases}$$

$\alpha = 10{,}2;\ \ a = 0{,}35\alpha.$

15. $$y = \begin{cases} \sin^2 \lambda - x^2, & x \le 0 \\ \cos\lambda + \sqrt{x^3 + 1}, & x < 0 \end{cases}$$

$\lambda = 0{,}35;\ \ x = \lambda^2 + a^2;\ a = 10.$

3. Алгоритми циклічної структури

Скласти блок-схеми алгоритмів обчислення значення функції Y за варіантами:

1.	$Z=\begin{cases}1+2{,}5e^{-0.8x}, & x>1\\ \sqrt{x^2+2y^2}, & x\le 1\end{cases}$ $x=a\sin\alpha;\ y=5.6-2x;\ a=8{,}3;\ \alpha_п=0{,}25;\ \alpha_к=0{,}50;\ \Delta\alpha=0{,}05.$
2.	$Z=\begin{cases}8{,}6\ln 3.5(1+x^2), & x<1{,}5\\ \sqrt[3]{4{,}8+y^2}, & x\ge 1{,}5\end{cases}$ $x=e^{1{,}5\alpha};\ y=ax^2-3{,}6;\ a=2{,}8,\ \alpha_п=0{,}35;\ \alpha_К=0{,}75;\ \Delta\alpha=0{,}05.$
3.	$Z=\begin{cases}7{,}2x^2\mathrm{tg}\alpha, & y>1\\ (1+2y^2)\cos\alpha, & y\le 1\end{cases}$ $x=a\cdot e^{3\alpha};\ y=2x\cdot\cos\alpha;\ a=3{,}7,\ \alpha_п=0{,}30;\ \alpha_к=0{,}60;\ \Delta\alpha=0{,}05.$
4.	$Z=\begin{cases}7{,}5x^2+2{,}8y^2, & x<3\\ 4.5-\sqrt{x^2+3.4y^2}, & x\ge 3\end{cases}$ $x=a\sin\alpha;\ y=x^2\cos\alpha;\ a=4{,}5;\ \alpha_п=0{,}20;\ \alpha_к=0{,}70;\ \Delta\alpha=0{,}10.$
5.	$Z=\begin{cases}\sqrt{6{,}7x^2+2y^2}, & x<1\\ 3{,}8e^{-0.6x}, & x\ge 1\end{cases}$ $x=1+2{,}7\mathrm{tg}\alpha;\ y=a(1+3{,}5\cos 2\alpha);\ a=0{,}30;\ \alpha_п=0{,}30;\ \alpha_к=0{,}60;\ \Delta\alpha=0{,}05.$
6.	$Z=\begin{cases}\sqrt{2{,}5y+x^2}, & x<2\\ 6.7\ln\lvert y\rvert, & x\ge 2\end{cases}$ $x=t\mathrm{tg}\alpha;\ y=\sqrt{\dfrac{x^2}{\cos\alpha}};\ t=6{,}75;\ \alpha_п=0{,}25;\ \alpha_к=0{,}48;\ \Delta\alpha=0{,}03.$
7.	$Z=\begin{cases}x^2\sin\alpha+y^2, & y>1\\ \sqrt{x^2+y^2}\mathrm{tg}\alpha, & y\le 1\end{cases}$ $x=a^2e^{1.5\alpha};\ y=3.8\ln 5\alpha;\ a=3{,}4;\ \alpha_п=0{,}30;\ \alpha_к=0{,}60;\ \Delta\alpha=0{,}05.$
8.	$Z=\begin{cases}3.5+\sqrt[3]{16.8+8.4\cdot x^2}, & x<2\\ (3.5+xy)e^x, & x\ge 2\end{cases}$ $x=(1+\mathrm{tg}^2\alpha)\cdot a;\ y=2{,}5x^2-6.7;\ \alpha_п=0{,}30;\ \alpha_к=0{,}80;\ \Delta\alpha=0{,}05;\ a=2{,}3.$

9.	$Z=\begin{cases}7{,}8e^{x}\cdot\sin\alpha, & y<1\\ \sqrt{1{,}6+x^{2}}\cdot y, & y\geq 1\end{cases}$ $x=(3{,}7-a^{2})\cos\alpha;\ y=a\sin\alpha;\ \alpha_{п}=0{,}40;\ \alpha_{к}=0{,}90;\ \Delta\alpha=0{,}05;\ a=7{,}6.$
10.	$Z=\begin{cases}\dfrac{(x^{2}+y^{2})}{xy}, & x<1{,}5\\ (x^{2}-y^{2})e^{xy}, & x\geq 1{,}5\end{cases}$ $x=8{,}6\sin\alpha\quad y=3{,}5tx\cos\alpha;\ \alpha_{п}=0{,}35;\ \alpha_{к}=0{,}75;\ \Delta\alpha=0{,}05;\ t=1{,}9.$
11.	$Z=\begin{cases}\sqrt{x^{2}+y^{2}e^{2x}}, & x>1\\ 1{,}7xy, & x\leq 1\end{cases}$ $x=a*tg(\alpha)^{2};\ y=(1+x)*\cos(2\alpha);\ \alpha_{п}=0{,}25;\ \alpha_{к}=0{,}50;\ \Delta\alpha=0{,}05;\ a=9{,}2.$
12.	$Z=\begin{cases}\sqrt{1+xy}, & x<1\\ (x^{2}+y^{2})e^{xy}, & x\geq 1\end{cases}$ $x=a\sin\alpha;\ y=3{,}8a+\sqrt{atg\alpha};\ \alpha_{п}=0{,}35;\ \alpha_{к}=0{,}75;\ \Delta\alpha=0{,}05;\ a=7{,}3.$
13.	$Z=\begin{cases}x\sin\alpha+y\cos\alpha, & y>2\\ ye^{x}, & y\leq 2\end{cases}$ $x=\sqrt{2{,}5+a^{2}tg\alpha};\ \ y=a\ln\alpha;\ \alpha_{п}=0{,}10;\ \alpha_{к}=0{,}60;\ \Delta\alpha=0{,}1;\ a=5{,}8.$
14.	$Z=\begin{cases}x^{2}\sin\alpha+y^{2}\cos\alpha, & x>2\\ \sqrt{xy+3{,}6tg^{\alpha}}, & x\leq 2\end{cases}$ $x=e^{2{,}5a};\ y=1+a^{2}x^{2};\ \alpha_{п}=0{,}14;\ \alpha_{к}=0{,}32;\ \Delta\alpha=0{,}04;\ a=18{,}6.$
15.	$Z=\begin{cases}\left(x^{2}-y^{2}\right)e^{x}, & y>1\\ 4{,}6+x^{2}y^{2}\sin\alpha, & y\leq 1\end{cases}$ $x=7{,}6\ln a;\quad y=tg^{2}\alpha;\ \alpha_{п}=0{,}35;\ \alpha_{к}=0{,}6;\ \Delta\alpha=0{,}05;\ a=1{,}4.$

10.4. Одномірні масиви

Скласти блок-схеми алгоритмів за варіантами:

Номер варіанта	Знайти:	Масив
1	Середнє арифметичне S додатних елементів і їхню кількість К.	-45,9; 2,82; 30,317; -43,5; 34,2; 13,2; 0; -10,4; 30,874
2	Суму S1 елементів із парними номерами і суму S2 елементів із непарними номерами.	17,76; -12,7 ;-16,8; 14,16; 36,325; 9,547; 15,796; -19,301; 0; 5,3; 6.8; -17,3
3	Суму S квадратів усіх елементів, що перевищують 10 по абсолютному значенню і їхню кількість К.	42,027; 0;-23,018; 0; -18,532; 0,73; 30,8; 39,115; 7,3;-18,67; 12,32;-8,05;-16,3
4	Добуток PI усіх додатних елементів і їхню кількість К1, а також, добуток Р2 усіх від'ємних елементів і їхню кількість К2.	43,175; -11,082; 0; 32,217; -5,42; -2,477; 13,921; -14,184; -7,3; 8,13; 16,08; 123,3; 18,67
5	Середнє арифметичне S квадратів усіх елементів, що перевищують 2,5 по абсолютному значенню та їхню кількість К.	-10,396; -3,47; -14,748; 0; -2,34; 43,796; -2,616; 46,139; 0,35; 5,75; -1,308; 7,87
6	Добуток відмінних від нуля елементів із непарними номерами і їхню кількість К1.	16,375; -17,004; -49,399; -43,353; -15,530; -3,001; 21,762; -42,420; 7,375; -0,675; 13,834; -7,68
7	Суму S квадратів елементів, значення яких належать відрізку (-10,12) і їхню кількість К.	-13,27; -9,547; 0; -22,477; 43,796; -3,001; -28,706; 9,488; -41,66; 13,879; 16,713

8	Середнє арифметичне S елементів, що не перевищують 15 по абсолютному значенню і їхню кількість К.	-28,221; 2,829; -18,7; -12,784; 0; -34,719; -17,04; -12,784; 1,89; 5,83; 56,13; -14,8
9	Значення М найбільшого элемента і його номер.	49,624; -20,481; 87,68; 0; 32,646; 118,37; 18,34; 5,68
10	Кількість N від'ємних, кількість P додатних і кількість Z нульових елементів.	11,749; 0; 0;-39,144; 0; 9,488; 21,412; 41,643; 5,6; 7,8; -0,34; -1,2; 17,13
11	Середнє арифметичне елементів, значення яких належать інтервалу (-273; 20), і їхню кількість.	206,8; -31,18; 0; 36,9; -313,8; 0,67; -230,2; 0; 21,18; 5,64; 115,36; -270.3, 18,8; 35,7; 6,4
12	Значення M найбільшого по абсолютному значенню елемента і його номер N.	-41,5; -22,174; 40,464; 42,347; -10,089; -41,66; 40,843; 0; 20; -47,591; 34,458; 7,83
13	Суму S квадратів від'ємних елементів з номерами, кратними трьом, і кількість додатних елементів.	35,066,0,276; -13,94; 13,879; 8,73, 0; -13,762;-29,777; 45,194;-25,613; 38,642
14	Середнє арифметичне S елементів, відмінних від нуля і їхню кількість К1, а також кількість КО елементів, рівних нулю.	-34,22; 36,325; -18,532; -5,42; 0; -23,401; -15,53; 0; 0;-0,089; -13,94; 0; 0;-13,914
15	Добуток P відмінних від нуля елементів і кількість елементів рівних нулю.	-10,423;-19,301; 39,115; 0; 4,184; 0; 0; 46,139; 42,42; 0; 20; 0;0

10.5. Двомірні масиви

Скласти блок-схеми алгоритмів за варіантами:

1. Знайти максимальній по значенню елемент в кожному рядку заданого двомірного масиву A(3,4).
2. Знайти кількість від'ємних елементів у кожному рядку заданого двомірного масиву B(4,3).
3. Знайти суму від'ємних елементів у кожному стовпчику заданого двомірного масиву C(2,3).
4. Знайти мінімальний елемент заданого двомірного масиву T(4,3). Вивести його значення та індекси
5. Знайти елементи двомірного масиву C(2,2), кожний з яких дорівнює сумі відповідних елементів заданих двомірних масивів A(2,2) і B(2,2).
6. Для кожного стовпчика заданого двомірного масиву K(4,4) знайти суму елементів , розташованих нижче головної діагоналі.
7. Для кожного рядка заданого двомірного масиву AH(4,4) знайти номера стовпчиків, які мають не нулеві елементи.
8. Поділити елементи кожного стовпчика заданого двомірного масиву CP(2,3) на останній елемент стовпчика.
9. Для кожного стовпчика заданого двомірного масиву T(3,4) знайти елемент, значення якого максимально.
10. Для кожного рядка заданого двомірного масиву P(4,3) знайти суму елементів стовпчиків з парними номерами.
11. Сформувати двомірний масив, кожний елемент якого є ціла частка відповідного елемента заданого двомірного масиву M(4,3).
12. Знайти мінімальній елемент у кожному стовпчику заданого двомірного масиву BP(3,4).
13. Знайти кількість додатних елементів у кожному стовпчику заданого двомірного масиву C(3,4).

14. Знайти суму елементів більших, ніж число C=2.5 для кожного рядка заданого двомірного масиву A(4,3).

15. Знайти максимальний елемент заданого двомірного масиву B(2,4). Вивести його значення та індекси.

11. Завдання підвищеної складності

1. Двомірний масив A(3,3) перетворити в одномірний, виконати сортування по зростанню. Всі три масиви вивести. Скласти блок-схему алгоритму.

2. У одномірному масиві змінити місцями 1-ий елемент з останнім, 2-ий − з передостаннім і так далі. Скласти блок-схему алгоритму.

3. Задано одномірний масив A. Створити два масиви, в один з яких записати п'ять перших додатних елементів масиву A, а у другий − номера цих елементів. Скласти блок-схему алгоритму.

4. Перевірити, чи є в заданому масиві хоча б один елемент, рівний або кратний своєму номеру в масиві. Скласти блок-схему алгоритму.

Література

1. Кормэн Т.Х., Лейзерсон Ч.И., Ривест Р.Л. Алгоритмы: построение и анализ − М.: МЦНМО, 2000. − 960 с.
2. Єжова Л. Ф. Алгоритмізація і програмування процедур обробки інформації: Навч.- метод. посібник для самост. вивч. дисц. — К.: КНЕУ, 2000. − 152 с.
3. Колдаев В.Д. Основы алгоритмизации и программирование: Учеб. пособие. − М.: ИД «ИНФРА-М, 2006. − 120 с.
4. Голицина О.Л., Попов И.И. Основы алгоритмизации и программирование − М.: ФОРУМ: ИНФРА-М. 2004. − 154 с.

ЗМІСТ

Навчальне видання

Швачич Генадій Григорович

Гуляєва Олена Анатоліївна

Оржех Олександр Ігоревич

АЛГОРИТМІЗАЦІЯ У ПРИКЛАДАХ ТА ЗАВДАННЯХ

Навчальний посібник

Тем. план 2010, поз. 196

Підписано до друку 19.04.2010. Формат 60х48 1/16. Папір друк. Друк плоский.
Облік. – вид. арк. 2,58. Умов. друк. арк. 2,56. Тираж 100 пр. Замовлення №

Національна металургійна академія України
49600, м. Дніпропетровськ–5, пр.. Гагаріна 4

Редакційно-видавничий відділ НМетАУ