

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

Добровольський Ю.М., Єфіменко К.М.

**Конспект лекцій за курсом
«Інформатика і основи програмування»**

УДК 004.432.2

Конспект лекцій за курсом «Інформатика і основи програмування»/
Ю.М. Добровольський, К.М. Єфіменко. – Донецьк:ДВНЗ «ДонНТУ». – 2010. –
92 с.

Наведено короткий базовий конспект лекцій за курсом «Інформатика і основи програмування», що читається для студентів I курсу технічних спеціальностей ДВНЗ «ДонНТУ».

Автори:

Ю.М. Добровольський,
К.М. Єфіменко.

Відп. за випуск:

В.М. Павлиш, д.т.н., професор.

© К.М. Єфіменко. 2010

© ДВНЗ «ДонНТУ». 2010

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

Добровольський Ю.М., Єфіменко К.М.

**Конспект лекцій за курсом
«Інформатика і основи програмування»**

Розглянуто
на засіданні кафедри ОМіП
протокол №9 від “6” квітня 2010 р.

Затверджено
навчально-видавничою радою ДонНТУ
протокол №2 від “22” квітня 2010 р.

ЗМІСТ

Ціль і задачі курсу.....	5
Лекція №1.	
Системи числення.....	6
Лекція №2.	
Поняття алгоритму. Алгоритмізація обчислювальних процесів.....	15
Лекція №3.	
Організація алгоритмів циклічної структури.....	20
Лекція №4.	
Алгоритми циклічної структури з невідомим числом повторень.....	24
Лекція №5.	
Організація вкладених циклів.....	26
Лекція №6.	
Організація ітераційного процесу.....	29
Лекція №7.	
Поняття індексованої змінної. Масиви.....	33
Лекція №8.	
Приклади обробки одномірних масивів.....	39
Лекція №9.	
Обробка двовимірних масивів.....	44
Лекція №10.	
Основи роботи в MS Excel.....	50
Лекція №11.	
Загальні відомості про об'єктно-орієнтованому програмуванні.....	58
Лекція №12.	
Умовний оператор. Оператори циклу в VBA.....	64
Лекція №13.	
Арифметичні оператори. Стандартні функції в VBA. Створення модулів. Процедури й функції користувача.....	68
Лекція №14.	
Оператори вводу-вивіду в VBA. Об'єкти, властивості й методи в VBA... Лекція №15.	73
Лекція №15.	
Приклади складання функції користувача.....	78
Лекція №16.	
Опис масивів в VBA. Основні властивості форм. Основні події й методи форм.....	81
Лекція №17.	
Методи обробки даних. Інтерполяція.....	86
Список літератури.....	91

Ціль і задачі курсу

Ціль курсу «Інформатика і основи програмування» – формування у студентів навичок алгоритмічного мислення, уміння здійснювати постановку задачі для розробки програмного забезпечення і реалізації алгоритмів у вигляді комп'ютерних програм.

Задачі курсу – вивчення організації обчислювальних процесів, принципів алгоритмізації, основних типів алгоритмів, способів їхнього представлення, освоєння етапів розробки програм.

У результаті вивчення курсу студент повинен:

- знати типи алгоритмів і етапи розробки програм;
- уміти розробляти алгоритми і програми для розв'язання задач на комп'ютері.

Курс складається з восьми лабораторних робіт. Кожна лабораторна робота виконується в наступному порядку:

1. Одержання індивідуального завдання відповідно до варіанта студента.
2. Розв'язання поставленої задачі.
3. Оформлення звіту.
4. Подання звіту викладачу.

Звіт по лабораторній роботі повинен містити наступні пункти:

1. Вхідні дані.
2. Постановка задачі (математична модель).
3. Обмеження на розв'язання задачі.
4. Вихідні дані.
5. Блок-схема алгоритму.
6. Короткий опис блок-схеми.
7. Текст програми розв'язання задачі.

Для програмної реалізації алгоритмів пропонується використати мову Visual Basic for Application (VBA), убудовану у додатки пакета Microsoft Office і яка поширює їх стандартні можливості.

Лекція №1. СИСТЕМИ ЧИСЛЕННЯ

1.1. Позиційні системи числення

Із древніх часів людство користується числами й операціями над ними. При цьому найбільше розповсюдження в повсякденному житті отримала десяткова система числення (с/ч), у якій використовуються десять арабських цифр від 0 до 9. Розташовуючи цифри в різних позиціях, ми отримуємо різні числа. Така с/ч називається позиційною, у ній величина числа визначається положенням і значенням кожної його цифри. Іншим прикладом позиційної с/ч може служити римська система числення. Поряд з позиційними існують і непозиційні системи числення. Наприклад, такою с/ч користувався Робінзон Крузо, що за допомогою карбів відзначав кількість проведених на незаселеному острові днів.

Будь-яке число N у позиційній системі числення можна представити в наступному вигляді:

$$N = (b_m b_{m-1} \dots b_2 b_1 b_0, b_{-1} b_{-2} \dots b_{-k})_A, \quad (1)$$

де b – цифра з алфавіту с/ч;

, – роздільник цілої й дробової частини числа;

A – основа системи числення (для десяткової с/ч $A = 10$);

$m \dots k$ – вагарні коефіцієнти.

Наприклад, десяткове число $205,74_{10}$ відповідно до (1) має наступні вагові коефіцієнти: $2^2 0^1 5^0, 7^{-1} 4^{-2}$.

Використовуючи ті ж позначення, число N можна представити у вигляді суми елементів ряду:

$$N = b_m \cdot A^m + b_{m-1} \cdot A^{m-1} + \dots + b_1 \cdot A^1 + b_0 \cdot A^0 + b_{-1} \cdot A^{-1} + \dots + b_{-k} \cdot A^{-k} \dots \quad (2)$$

Наприклад: $205,74_{10} \rightarrow 2^2 0^1 5^0, 7^{-1} 4^{-2} \rightarrow 2 \cdot 10^2 + 0 \cdot 10^1 + 5 \cdot 10^0 + 7 \cdot 10^{-1} + 4 \cdot 10^{-2}$

Наведений вираз (2), є універсальним для будь-якої позиційної системи числення.

Незважаючи на те, що десяткова с/ч є для нас найбільш звичною й зручною у використанні, реалізація на її основі обчислювальної техніки не раціонально. Тому основною системою числення для внутрішнього зберігання й оброб-

ки даних у комп'ютері є **двійкова** й похідні від неї восьмерична й шістнадцятерична с/ч.

1.1.1. Двійкова система числення

У двійковій системі числення (2 с/ч) використовується дві цифри 0 і 1, основа с/ч $A=2$. Наприклад, двійкове число 101101_2 відповідає десятковому числу 45_{10} .

Використання у двійковій с/ч мінімальної кількості цифр, для запису чисел, дозволяє найбільше економічно реалізовувати апаратну частину ЕОМ. Кожна цифра двійкового числа називається **біт**. Біт називається також двійковим розрядом. Група з 8 біт складає **байт**, що може зберігати різні типи даних, такі як літери алфавіту, десяткові цифри або інші знаки. Таким чином, **1 біт = 2^{-3} байт**.

Байт є основною одиницею виміру інформації. Крім цього для виміру обсягу інформації часто використовуються наступні похідні від байта:

$$1 \text{ Кбайт (кілобайт)} = 1024 \text{ байт} = 2^{10} \text{ байт},$$

$$1 \text{ Мбайт (мегабайт)} = 1024 \text{ Кбайт} = 2^{20} \text{ байт},$$

$$1 \text{ Гбайт (гігабайт)} = 1024 \text{ Мбайт} = 2^{30} \text{ байт},$$

$$1 \text{ Тбайт (терабайт)} = 1024 \text{ Гбайт} = 2^{40} \text{ байт}.$$

Для конвертування чисел із двійкової с/ч у десяткову використовуються вирази (1, 2).

Приклад. $11101,01_2 \rightarrow (?)_{10}$

Послідовність вагових коефіцієнтів має вигляд

$$1^4 1^3 1^2 0^1 1^0, 0^{-1} 1^{-2} \rightarrow 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 29,25_{10}$$

Однак істотним недоліком двійкової с/ч є громіздкий запис чисел. Для спрощення запису двійкових чисел можуть бути використані восьмерична або шістнадцятерична системи числення.

1.1.2. Восьмерична система числення

У восьмеричній системі числення, що є похідною від двійкової, використовується вісім цифр від 0 до 7, і її основа $A=8$. Основу восьмеричної с/ч, тобто число 8, можна представити у вигляді 2^3 . Тому одній восьмеричній цифрі відповідає три двійкових розряди – **тріада**.

Відповідність між восьмеричним числом і його двійковим і десятковим представленнями наведено в таблиці 1.1.

Таблиця 1.1

Зв'язок між восьмеричною, двійковою й десятковою с/ч

Восьмерична с/ч	Двійкова с/ч	Десяткова с/ч
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
10	001000	8
11	001001	9
12	001010	10

Для конвертування двійкового числа у восьмеричне необхідно розбити двійкове число на тріади в такий спосіб: ціла частина розбивається на тріади, починаючи з молодших розрядів (із правого краю цілої частини числа), а дробова частина - з лівого краю. Розряди, яких не вистачає для формування тріад з лівого або правого країв, заповнюються 0. Отримані тріади за допомогою таблиці 1.1, замінюються восьмеричними цифрами.

Приклад. $1011010110,01_2 \rightarrow (?)_8$

$1011010110,01_2 \rightarrow \underline{001} \underline{011} \underline{010} \underline{110}, \underline{010} \rightarrow 1326,2_8$

Для виконання зворотного переведення (з 8 с/ч в 2 с/ч) кожен восьмеричну цифру заміняють відповідною двійковою тріадою. Незначні 0 у цілій і дробовій частинах отриманого числа можна відкинути.

Приклад. $357,24_8 \rightarrow (?)_2$

$357,24_8 \rightarrow \underline{011} \underline{101} \underline{111}, \underline{010} \underline{100} \rightarrow 11101111,0101_2$

Для конвертування числа з восьмеричної с/ч у десяткову також використовуються вирази (1, 2).

Приклад. $357,24_8 \rightarrow (?)_{10}$

послідовність вагових коефіцієнтів має вигляд $3^2 5^1 7^0, 2^{-1} 4^{-2} \rightarrow$

$$3 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 + 2 \cdot 8^{-1} + 4 \cdot 8^{-2} = 239,3125_{10}$$

1.1.3. Шістнадцятерична система числення

У шістнадцятеричній системі числення використовується десять цифр від 0 до 9 і шість латинських літер A, B, C, D, E, F. Основу с/ч $A=16$ можна представити у вигляді 2^4 . Тому однієї шістнадцятеричній цифрі відповідає чотири двійкових розряди – **тетрада**.

Відповідність між шістнадцятеричним числом і його двійковим і десятковим представленнями наведено в таблиці 1.2.

Конвертування із двійкової с/ч у шістнадцятеричну с/ч і зворотно виконується способом, описаним для восьмеричної с/ч. Отримані при цьому тетради за допомогою таблиці 1.2, заміняються шістнадцятеричними цифрами й навпаки.

Приклад: $1011010110,01_2 \rightarrow (?)_{16}$

$1011010110,01_2 \rightarrow \underline{0010} \underline{1101} \underline{0110}, \underline{0100} \rightarrow 2D6,2_{16}$

$4A9,B_{16} \rightarrow (?)_2$

$4A9,B_{16} \rightarrow \underline{0100} \underline{1010} \underline{1001}, \underline{1011} \underline{0010} \rightarrow 10010101001,1011001_2$

При конвертуванні із шістнадцятеричної с/ч у десяткову також використо-

вуються вирази (1, 2).

Таблиця 1.2

Зв'язок між шістнадцятеричною, двійковою й десятковою с/ч.

Шістнадцятерична с/ч	Двійкова с/ч	Десяткова с/ч	Шістнадцятерична с/ч	Двійкова с/ч	Десяткова с/ч
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	A	1010	10
3	0011	3	B	1011	11
4	0100	4	C	1100	12
5	0101	5	D	1101	13
6	0110	6	E	1110	14
7	0111	7	F	1111	15
			10	00010000	16

Приклад: $2D6,2_{16} \rightarrow (?)_{10}$

послідовність вагових коефіцієнтів має вигляд $2^2 D^1 6^0, 2^{-1} \rightarrow$

$$2 \cdot 16^2 + 13 \cdot 16^1 + 6 \cdot 16^0 + 2 \cdot 16^{-1} = 726,125_{10}$$

Переведення із шістнадцятеричної с/ч у восьмеричну й навпаки виконується у два етапи. На початку виконується конвертування у двійкову с/ч, а потім із двійкової в необхідну с/ч, описаним вище способом.

Приклад: $4A9, B_{16} \rightarrow (?)_8$

$$4A9, B_{16} \rightarrow \underline{0100} \underline{1010} \underline{1001}, \underline{1011}_2 \rightarrow \underline{010} \underline{010} \underline{101} \underline{001}, \underline{101} \underline{100}_2 \rightarrow 2251,54_8$$

$$357,24_8 \rightarrow (?)_{16}$$

$$357,24_8 \rightarrow \underline{011} \underline{101} \underline{111}, \underline{010} \underline{100}_2 \rightarrow \underline{0000} \underline{1110} \underline{1111}, \underline{0101} \underline{0000}_2 \rightarrow EF,5_{16}$$

1.2. Переведення цілої частини десяткового числа у різні системи числення

Переведення цілого числа представленого в десятковій с/ч у двійкову, во-

сьмеричну або шістнадцятеричну системи числення виконується шляхом цілочисельного ділення вхідного числа на основу нової с/ч. При цьому необхідно виконати наступну послідовність дій:

1. Розділити націло вхідне число на основу нової с/ч. Остача від ділення дає молодшу цифру числа в новій с/ч.
2. Якщо частка від ділення не дорівнює 0, то перейти до пункту 3, інакше перейти до пункту 4.
3. Розділити отриману частку націло на основу нової с/ч. Остача від ділення дає наступну цифру числа в новій с/ч. Перейти до пункту 2.
4. Отримані в результаті ділень остачі, записати в порядку зворотному їхньому обчисленню. Це й буде вхідне число в новій с/ч.

Приклад: $29_{10} \rightarrow (?)_2$

$$\begin{array}{r}
 29 \mid 2 \\
 \underline{28} \mid 14 \quad 2 \\
 1 \mid 14 \quad 7 \quad 2 \\
 \quad \underline{0} \mid 6 \quad 3 \quad 2 \\
 \quad \quad \underline{1} \mid 2 \quad 1 \quad 2 \\
 \quad \quad \quad \underline{1} \mid 0 \quad 0 \\
 \quad \quad \quad \quad \underline{1}
 \end{array}$$

$$29_{10} \rightarrow 11101_2$$

$117_{10} \rightarrow (?)_8$

$$\begin{array}{r}
 117 \mid 8 \\
 \underline{8} \mid 14 \quad 8 \\
 37 \mid 8 \quad 1 \quad 8 \\
 \underline{32} \mid 6 \quad 0 \quad 0 \\
 \quad \quad \underline{5} \quad \quad \underline{1}
 \end{array}$$

$$117_{10} \rightarrow 165_8$$

$332_{10} \rightarrow (?)_{16}$

$$\begin{array}{r}
 332 \mid 16 \\
 \underline{320} \mid 20 \quad 16 \\
 12 \mid 16 \quad 1 \quad 16 \\
 \quad \underline{4} \mid 0 \quad 0 \\
 \quad \quad \underline{1}
 \end{array}$$

$$332_{10} \rightarrow 14C_{16}$$

Особливу увагу варто приділити переведенню в шістнадцятеричну с/ч, тому що залишки від цілочисельного ділення в цьому випадку, можуть перевищувати число 9, тому при записі кінцевого результату, важливо не забувати замінити їх відповідною шістнадцятеричною цифрою (позначуваною латинськими літерами від А до F).

1.3. Переведення дробової частини десяткового числа в різні системи числення із заданою точністю

Переведення дробової частини числа представленого в десятковій с/ч у двійкову, восьмеричну або шістнадцятеричну системи числення виконується

шляхом множення дробової частини вхідного числа на основу нової с/ч. Тому що процес послідовного множення може тривати нескінченно, то переведення виконується або до одержання необхідної кількості розрядів у дробовій частині числа в новій с/ч або до досягнення заданої точності. При цьому необхідно виконати наступну послідовність дій:

1. Помножити дробову частину вхідного числа на основу нової с/ч. Ціла частина отриманого добутку дає першу цифру дробової частини числа в новій с/ч.

2. Дробову частину отриманого добутку помножити на основу нової с/ч. Ціла частина отриманого добутку дає наступну цифру дробової частини числа в новій с/ч.

3. Якщо досягнуто задану точність або отримано необхідну кількість цифр у дробовій частині числа в новій с/ч те перейти до п. 4, інакше повторити п. 2.

4. Отримані в результаті множення цілі частини добутків, записати в порядку їхнього обчислення. Це й буде дробова частина вхідного числа в новій с/ч.

Приклад. Виконати переведення дробової частини десяткового числа у двійкову, восьмеричну й шістнадцятіречну с/ч, як критерій використовуючи, необхідність одержати 3 цифри в дробовій частині числа в новій с/ч.

$0,74_{10} \rightarrow (?)_2$	$0,23_{10} \rightarrow (?)_8$	$0,12_{10} \rightarrow (?)_{16}$
$\begin{array}{r} 0,74 \\ \underline{\quad 2} \\ \downarrow \underline{1}, 48 \\ \underline{\quad 2} \\ \downarrow \underline{0}, 96 \\ \underline{\quad 2} \\ \downarrow \underline{1}, 92 \end{array}$	$\begin{array}{r} 0,23 \\ \underline{\quad 8} \\ \downarrow \underline{1}, 84 \\ \underline{\quad 8} \\ \downarrow \underline{6}, 72 \\ \underline{\quad 8} \\ \downarrow \underline{5}, 76 \end{array}$	$\begin{array}{r} 0,12 \\ \underline{\quad 16} \\ \downarrow \underline{1}, 92 \\ \underline{\quad 16} \\ \downarrow \underline{14}, 72 \\ \underline{\quad 16} \\ \downarrow \underline{11}, 52 \end{array}$

$$0,74_{10} \rightarrow 0,101_2$$

$$0,23_{10} \rightarrow 0,165_8$$

$$0,12_{10} \rightarrow 0,1EB_{16}$$

Переведемо отримані значення назад в десяткову систему числення й визначимо Δ_A погрішність даного способу переведення.

$$0,101_2 \rightarrow 0^0, 1^{-1} 0^{-2} 1^{-3} = 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0,625_{10}$$

$$\text{Погрішність для двійкової с/ч складе } \Delta_2 = 0,74 - 0,625 = 0,115.$$

$$0,165_8 \rightarrow 0^0, 1^{-1} 6^{-2} 5^{-3} = 0 \cdot 8^0 + 1 \cdot 8^{-1} + 6 \cdot 8^{-2} + 5 \cdot 8^{-3} \approx 0,229_{10}$$

Погрішність для восьмеричної с/ч складе $\Delta_8 = 0,23 - 0,229 = 0,001$.

$$0,1EB_{16} \rightarrow 0^0, 1^{-1} E^{-2} B^{-3} = 0 \cdot 16^0 + 1 \cdot 16^{-1} + 14 \cdot 16^{-2} + 11 \cdot 16^{-3} \approx 0,1199_{10}$$

Погрішність для шістнадцятеричної с/ч складе $\Delta_{16} = 0,12 - 0,1199 = 0,0001$.

Проаналізувавши отримані результати, можна зробити висновок: чим менше інформаційна ємність системи числення, тим більше погрішність переведення. Отже, використаний критерій переведення не придатний для виконання розрахунків.

Як відомо, **точність**, з якої задане число, визначається кількістю цифр у його дробовій частині.

Для десяткової с/ч точність представлення числа визначається в такий спосіб:

$$(0, b_{-1} b_{-2} \dots b_{-k})_{10} \rightarrow b_{-1} \cdot 10^{-1} + b_{-2} \cdot 10^{-2} + \dots + b_{-k} \cdot 10^{-k}$$

Звідси:

точність, що дає цифра з ваговим коефіцієнтом -1 дорівнює $10^{-1} = 0.1$;

точність, що дає цифра з ваговим коефіцієнтом -2 дорівнює $10^{-2} = 0.01$;

... ..

точність, що дає цифра з ваговим коефіцієнтом -k дорівнює 10^{-k} .

У загальному випадку, точність, з якої задається число, у будь-якій позиційній системі числення визначається з виразу

$$t_A = \frac{1}{A^R}, \quad (4)$$

де A – основа системи числення;

R – кількість цифр у дробовій частині числа.

Точність переведення задається в десятковій системі числення (t_{10}). Для виконання переведення із заданою точністю необхідно одержати таку кількість цифр у дробовій частині числа в новій системі числення A , щоб $t_{10} > t$.

Приклад. Виконати переведення $0,74_{10} \rightarrow (?)_2$ з точністю $t = 0,01$.

$$0,74 \quad 0,74_{10} \rightarrow 0,1^{-1} 0^{-2} 1^{-3} 1^{-4} 1^{-5} 1^{-6} 0^{-7} = 0,1011110_2$$

<u>1</u> , 48	точність, що дає ця цифра $t_2 = 2^{-1} = 0,5 > t = 0,01$;
<u>0</u> , 96	точність, що дає ця цифра $t_2 = 2^{-2} = 0,25 > t = 0,01$;
<u>1</u> , 92	точність, що дає ця цифра $t_2 = 2^{-3} = 0,125 > t = 0,01$;
<u>1</u> , 84	точність, що дає ця цифра $t_2 = 2^{-4} = 0,0625 > t = 0,01$;
<u>1</u> , 68	точність, що дає ця цифра $t_2 = 2^{-5} = 0,03125 > t = 0,01$;
<u>1</u> , 36	точність, що дає ця цифра $t_2 = 2^{-6} \approx 0,01563 > t = 0,01$;
<u>0</u> , 72	точність, що дає ця цифра $t_2 = 2^{-7} \approx 0,00781 < t = 0,01$;

Отримана точність представлення числа у двійковій с/ч t_2 менше заданої точності t_{10} , отже, процес переведення можна завершити. Визначимо погрішність переведення, попередньо виконавши зворотне переведення:

$$0,1011110_2 \rightarrow 0^0, 1^{-1} 0^{-2} 1^{-3} 1^{-4} 1^{-5} 1^{-6} 0^{-7} =$$

$$= 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} + 0 \cdot 2^{-7} \approx 0,734_{10}$$

Погрішність склала $\Delta_2 = 0,74 - 0,734 = 0,006$ і не перевищує t_{10} .

Приклад. Виконати переведення $0,23_{10} \rightarrow (?)_8$ з точністю $t = 0,001$.

$$0,23 \quad 0,23_{10} \rightarrow 0,1^{-1} 6^{-2} 5^{-3} 6^{-4} = 0,1656_8$$

<u>1</u> , 84	точність, що дає ця цифра $t_8 = 8^{-1} = 0,125 > t = 0,001$;
<u>6</u> , 72	точність, що дає ця цифра $t_8 = 8^{-2} = 0,015625 > t = 0,001$;
<u>5</u> , 76	точність, що дає ця цифра $t_8 = 8^{-3} \approx 0,00195 > t = 0,001$;
<u>6</u> , 08	точність, що дає ця цифра $t_8 = 8^{-4} \approx 0,00024 < t = 0,001$;

Отримана точність представлення числа у восьмеричній с/ч t_8 менше заданої точності t_{10} , отже, процес переведення можна завершити. Визначимо погрішність переведення, попередньо виконавши зворотне переведення:

$$0,1656_8 \rightarrow 0^0, 1^{-1} 6^{-2} 5^{-3} 6^{-4} = 0 \cdot 8^0 + 1 \cdot 8^{-1} + 6 \cdot 8^{-2} + 5 \cdot 8^{-3} + 6 \cdot 8^{-4} \approx 0,22998_{10}$$

Погрішність склала $\Delta_8 = 0,23 - 0,22998 = 0,00002$ і не перевищує t_{10} .

Переведення десяткового числа, що містить цілу й дробову частини, у двійкову, восьмеричну або шістнадцятеричну системи числення, відбувається у два етапи. Спочатку переводиться ціла частина, а потім дробова частина числа.

1.4. Завдання для самостійної роботи

Послідовно виконати переведення чисел у зазначені системи числення (переведення з десяткової с/ч виконати із заданою точністю):

- 1) $1011001,01011_2 \rightarrow (?)_{10} \rightarrow (?)_8$ (точність переведення $t = 0,002$) $\rightarrow (?)_2$.
- 2) $546,26_8 \rightarrow (?)_{10} \rightarrow (?)_2$ (точність переведення $t = 0,01$) $\rightarrow (?)_8$.
- 3) $1AE,38_{16} \rightarrow (?)_{10} \rightarrow (?)_8$ (точність переведення $t = 0,0005$) $\rightarrow (?)_2 \rightarrow (?)_{16}$.
- 4) $1101001101010.1001_2 \rightarrow (?)_8 \rightarrow (?)_{16} \rightarrow (?)_2$.
- 5) $1011101110.01011_2 \rightarrow (?)_{16} \rightarrow (?)_8 \rightarrow (?)_2$.
- 6) $63417,464_8 \rightarrow (?)_{16} \rightarrow (?)_8$.
- 7) $F09C,08D6_{16} \rightarrow (?)_8 \rightarrow (?)_{16}$.
- 8) $94,37_{10} \rightarrow (?)_2$ (точність переведення $t = 0,04$) $\rightarrow (?)_{16} \rightarrow (?)_{10}$.
- 9) $238,512_{10} \rightarrow (?)_8$ (точність переведення $t = 0,00025$) $\rightarrow (?)_2 \rightarrow (?)_{10}$.
- 10) $493,951_{10} \rightarrow (?)_{16}$ (точність переведення $t = 0,00001$) $\rightarrow (?)_8 \rightarrow (?)_{10}$.

Лекція №2. ПОНЯТТЯ АЛГОРИТМУ.

АЛГОРИТМІЗАЦІЯ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

Алгоритм – це точний і простий опис послідовності дій, що однозначно визначають процес обчислення результату вирішення задачі при заданих вхідних даних. При складанні алгоритму використовуються різні способи запису:

аналітичний, описовий, графічний.




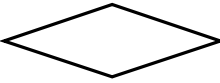

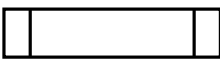


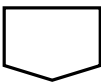
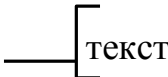
Самим наочним способом представлення алгоритму є графічний, у вигляді блок-схем, що складаються зі зв'язаних між собою блоків.

2.1. Типові структури алгоритмів

Графічні символи, які застосовуються при складанні блок-схем, представлені в таблиці 2.1.

Таблиця 2.1

Графічні символи, які застосовуються при складанні блок-схем

№ з/п	Найменування блоку	Позначення	Виконувана дія
1	Початок або кінець алгоритму		Показує початок або кінець алгоритму
2	Ввід або вивід		Забезпечує ввід або вивід даних в алгоритмі
3	Арифметичний		Виконує арифметичні обчислення
4	Логічний		Виконує перевірку заданої логічної умови
5	Модифікації		Заголовок циклу «Для»
6	Визначений процес		Виклик підпрограми
7	Лінії потоку		Указують зв'язок і напрямок руху між блоками
8	З'єднувач		Указує зв'язок між перерваними лініями потоку
9	Міжсторінковий з'єднувач		Указує зв'язок між частинами блок-схеми, які розташовані на різних сторінках
10	Коментарі		Запис пояснення до блоку або до лінії потоку

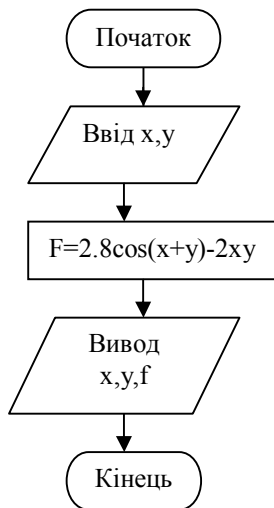
По характеру зв'язку між блоками (за структурою) алгоритми діляться на лінійні, що розгалужуються й циклічні.

Алгоритм лінійної структури. Це алгоритм, у якому блоки виконуються послідовно один за іншим у природному порядку.

Алгоритм структури, що розгалужується. Вважатиме алгоритмом, що розгалужується такий, якщо в ньому здійснюється вибір ходу розв'язання залежно від результату перевірки якої-небудь умови.

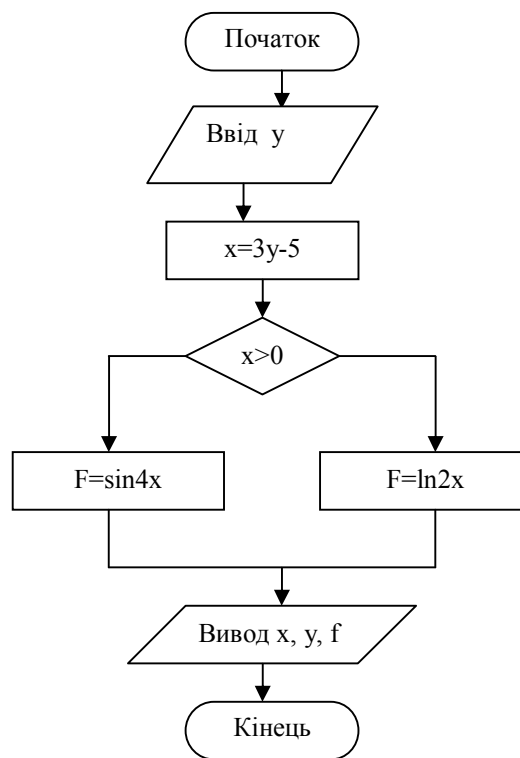
Приклад №1.

Обчислити $f=2.8\cos(x+y)-2xy$,
при $x=20.1$, $y=10.5$



Приклад №2.

Обчислити $f(x)=\begin{cases} \ln 2x, & \text{якщо } x>0 \\ \sin 4x, & \text{якщо } x\leq 0, \end{cases}$
де $x=3y-5$, $y=11.3$



2.2. Алгоритм циклічної структури

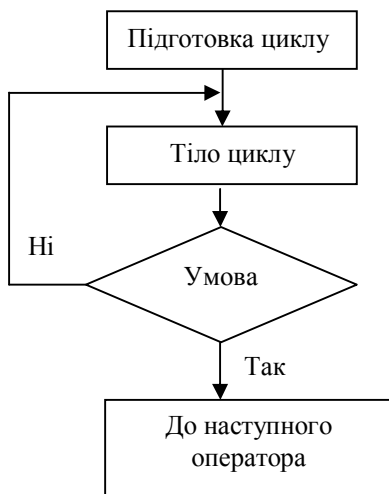
Це алгоритм, у якому міститься послідовність дій, що виконується неодноразово, яка називана циклом. Незалежна змінна, що змінюється в циклі, називається параметром циклу.

Цикли бувають із заданим числом повторень і з невідомим числом повторень. До циклів з невідомим числом повторень належать ітераційні (тобто при кожному наступному обчисленні області циклу результат обчислення наближається до шуканого із заданою точністю).

Можливі три способи організації циклічних структур:

- цикли із **пістумовою**;
- цикли із **передумовою**;
- цикли “Для”.

Цикл із пістумовою.



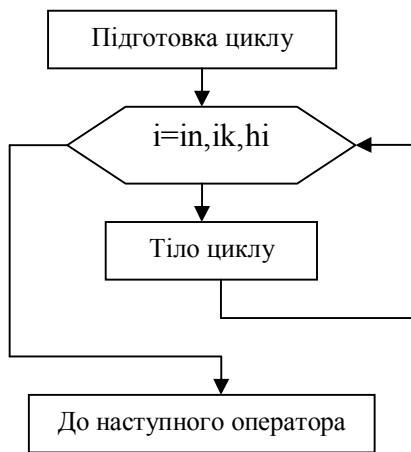
При такій організації циклу, тіло циклу виконується хоча б один раз. У тілі циклу повинен бути хоча б один блок, що впливає на умову, щоб уникнути зациклення. Спочатку виконується тіло циклу й перевіряється умова. Якщо умова виходу із циклу “неправда”, то тіло циклу повторюється. Якщо умова “істина”, то управління передається наступному за циклом оператору, тобто відбувається вихід із циклу.

Цикл із передумовою.



Перевіряється умова, якщо умова “Істина”, те тіло циклу виконується, у противному випадку управління передається наступному за циклом оператору, тобто здійснюється вихід із циклу. Якщо із самого початку умова “неправда”, то тіло циклу не буде виконане жодного разу. У тілі циклу повинен бути хоча б один блок, який впливає на умову, щоб уникнути зациклення.

Цикл «Для»



Параметром циклу є змінна i . Параметру i присвоюється початкове значення in і порівнюється з кінцевим значенням ik ($i=in$).

Якщо $i \leq ik$ (при $hi > 0$), то тіло циклу виконується.

Якщо $i > ik$ (при $hi > 0$), то здійснюється вихід із циклу.

Якщо $i \geq ik$ (при $hi < 0$), то тіло циклу виконується.

Якщо $i < ik$ (при $hi < 0$), то управління передається наступному за циклом оператору.

Після виконання тіла циклу управління передається блоку модифікації. У ньому до параметра циклу i додається крок hi і знову відбувається порівняння поточного значення i з кінцевим його значенням ik .

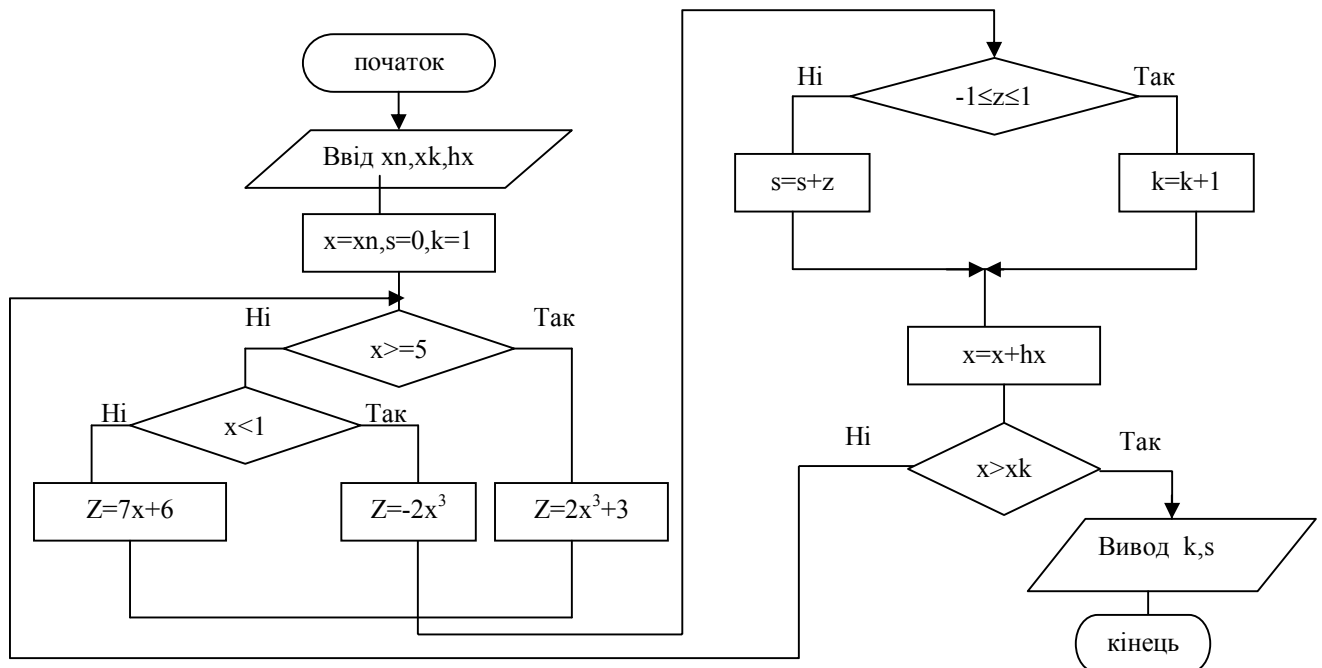
Цикл, до складу якого не входять інші цикли, називається простим, у протилежному випадку - складним. Цикл, що знаходиться усередині іншого циклу, є внутрішнім.

Приклад №3.

$$z = \begin{cases} 2x^3 + 3, & \text{якщо } x \geq 5 \\ 7x + 6, & \text{якщо } 1 \leq x < 5 \\ -2x^3, & \text{якщо } x < 1 \end{cases}$$

Вхідні дані
 $0 \leq x \leq 10$
 $hx = 1$

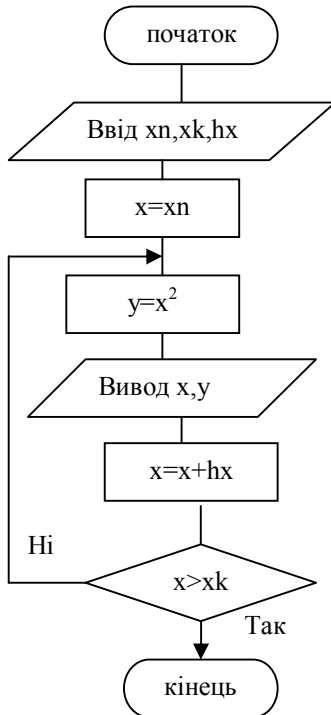
Виведені величини
 z, x
 кількість $z \in [-1; 1]$;
 сума $z \notin [-1; 1]$



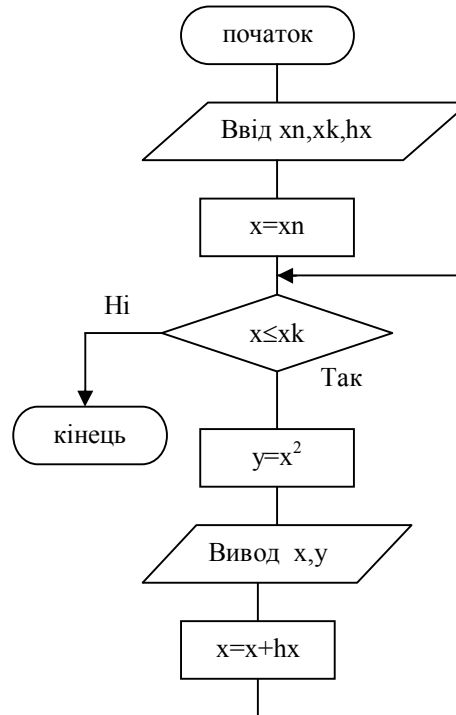
Лекція №3. ОРГАНІЗАЦІЯ АЛГОРИТМІВ ЦИКЛІЧНОЇ СТРУКТУРИ

Приклад №1. Обчислити значення функції $y=x^2$, при $1.2 \leq x \leq 2.6$, $hx=0.3$

Цикл із пістумовою.



Цикл із передумовою.



Цикл “Для”

Будемо позначати $]a[$ - ціла частина числа.

Визначення. Цілою частиною числа a називається найбільше ціле число не переважаюче дане. $]a[\leq a$.

Організувати цикл “Для” можна в такий спосіб: обчислити кількість x (повторень) до входження в цикл по формулі:

$$m = \left\lceil \frac{x_k - x_n}{h_x} \right\rceil + 1.$$

Потім, за допомогою блоку модифікації організувати цикл, де змінна управління циклом буде цілим числом: $\text{◇ } i=1,m$

Крок 1 у блоці можна не вказувати. Це обов'язково треба робити, якщо програмуємо на мові ПАСКАЛЬ (тому що в мові ПАСКАЛЬ змінна циклу повинна бути цілим числом, а крок дорівнює $(h=\pm 1)$).

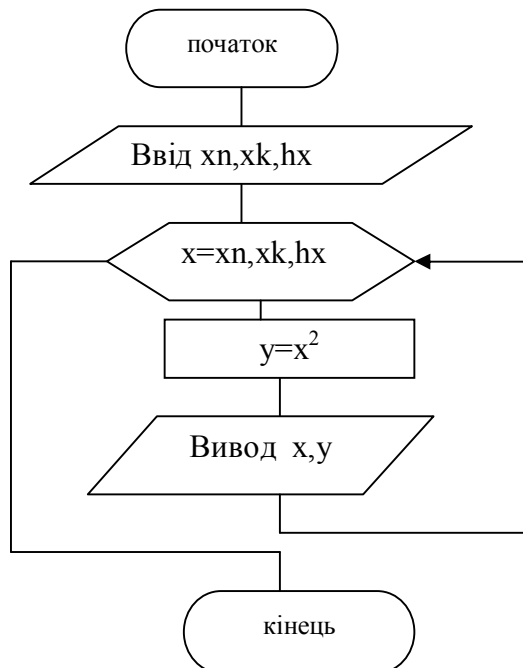
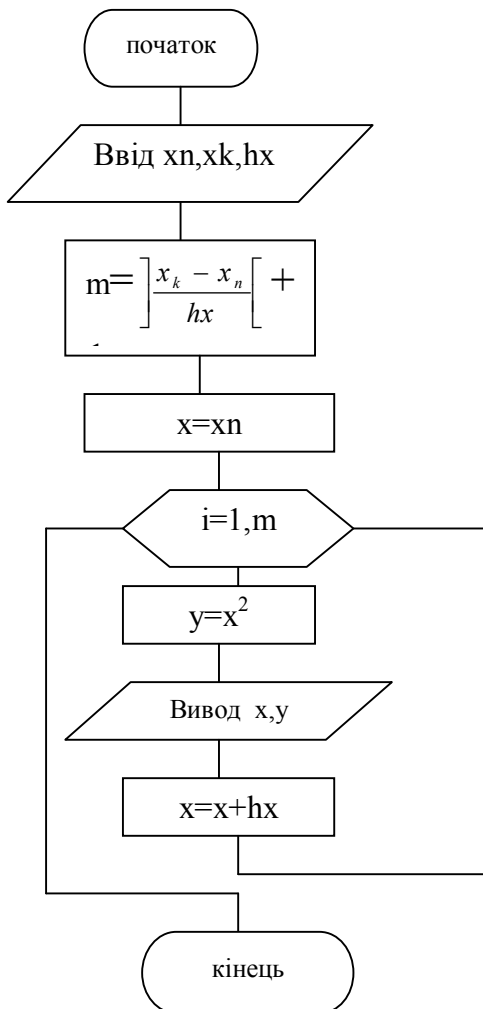
Якщо програмуємо на VBA, то це можна не робити, а просто записати:

$$x=xn,xk,hx$$

Повернемося до нашого прикладу:

$$m = \left\lceil \frac{x_k - x_n}{hx} \right\rceil + 1 = \left\lceil \frac{2,6 - 1,2}{0,3} \right\rceil + 1 = \left\lceil \frac{1,4}{0,3} \right\rceil + 1 = \left\lceil 4 \frac{2}{3} \right\rceil + 1 = 4 + 1 = 5$$

Справді, $x=1,2; 1,5; 1,8; 2,1; 2,4$.



Надалі (якщо не обговорено в умові) будемо позначати:

S - сума,

P - добуток,

k - кількість значень.

Наприклад $S = \sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n$ $S = \sum_{z \in [-1;1]}$ $P = \prod_{i=1}^m b_i = b_1 \cdot b_2 \cdot \dots \cdot b_m$, $P = \prod_{y>0}$

Лабораторна робота №2.

Тема: «Алгоритми циклічної структури з відомим числом повторень»

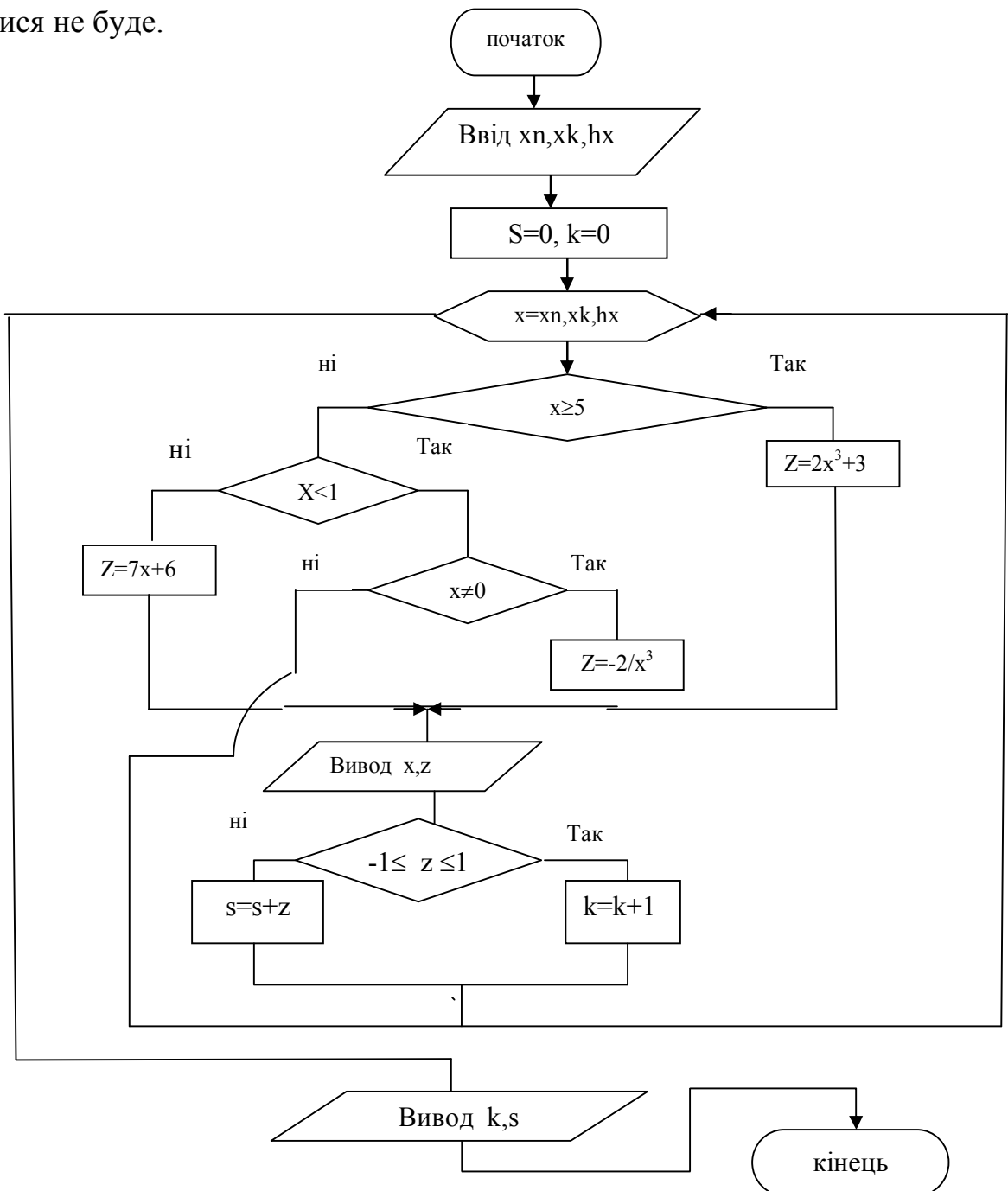
Приклад №1.

$$Z = \begin{cases} 2x^3 + 3, & \text{якщо } x \geq 5 \\ 7x + 6, & \text{якщо } 1 \leq x < 5 \\ -2/x^3, & \text{якщо } x < 1 \end{cases}$$

Вхідні дані: $0 \leq x \leq 10, hx = 1$
 Вивод.дані: z, x
 а також кіл – сть $Z \in [-1, 1]$

$$S = \sum_{z \in [-1; 1]} Z.$$

Організувати цикл із **підумовою** або із **передумовою**, із циклом «Для» робота прийматися не буде.

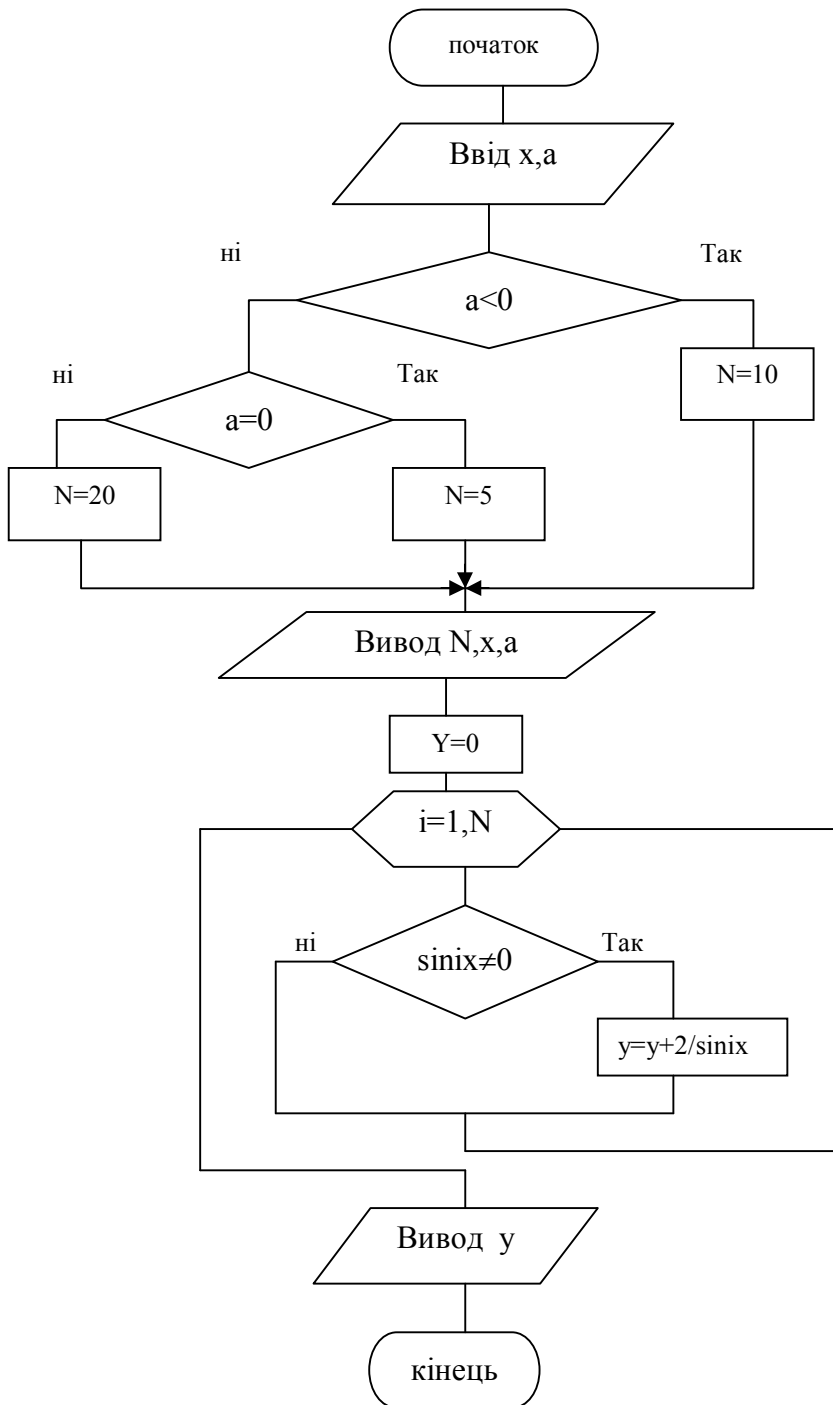


Приклад №2.

$$\text{Обчислити: } Y = \sum_{i=1}^N \frac{2}{\sin ix} \quad N = \begin{cases} 10, & \text{если } a < 0 \\ 5, & \text{если } a = 0 \\ 20, & \text{если } a > 0 \end{cases}$$

Вхідні дані: X=0.8; a

Виведені дані: y, N, a, x



Лекція №4. АЛГОРИТМИ ЦИКЛІЧНОЇ СТРУКТУРИ З НЕВІДОМИМ ЧИСЛОМ ПОВТОРЕНЬ

4.1. Лабораторна робота №3.

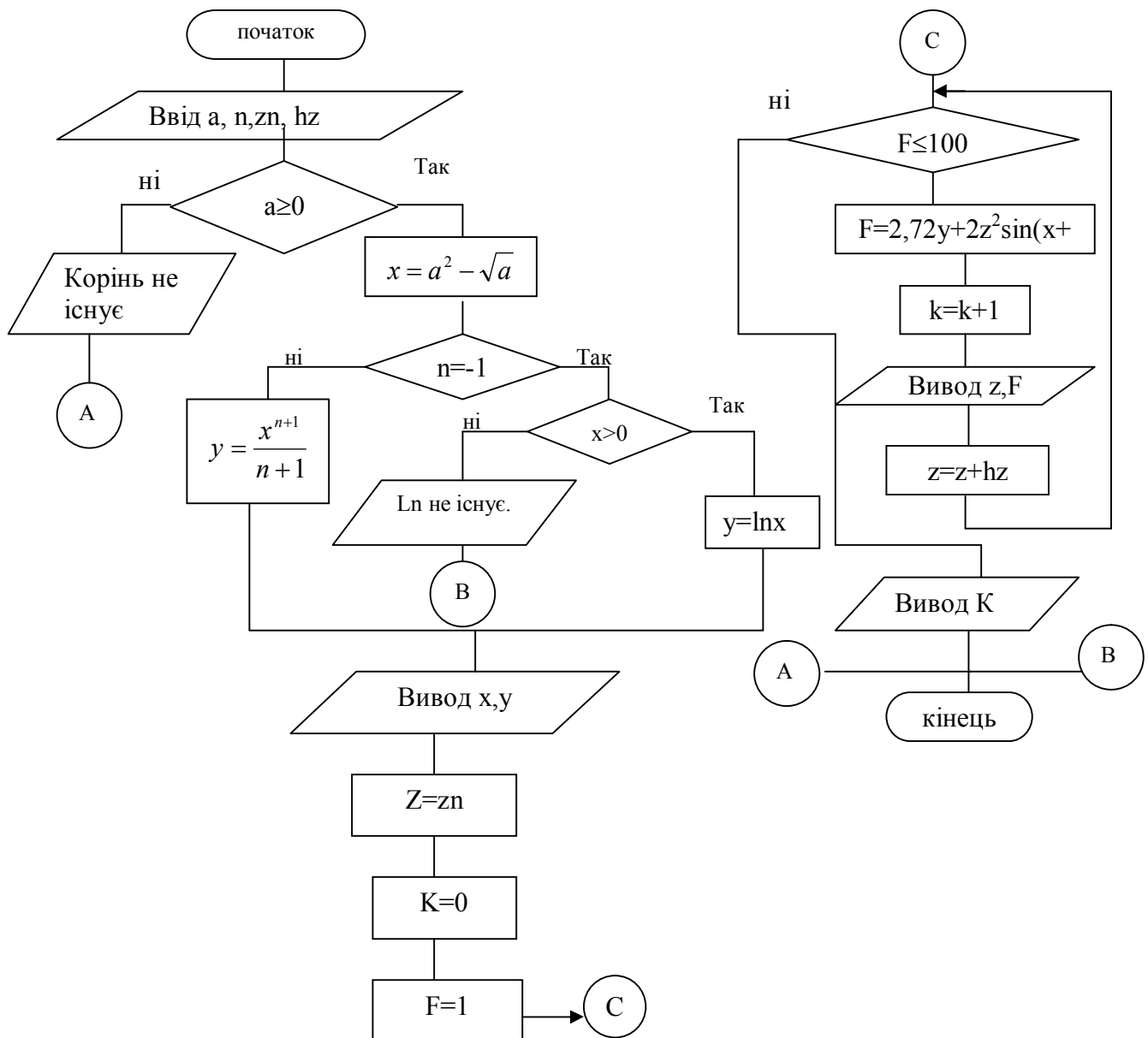
Постановка задачі:

$$F=2,72y+2z^2\sin(x+y), \quad x = a^2 - \sqrt{a}, \quad y = \begin{cases} \frac{x^{n+1}}{n+1}, & \text{якщо } n \neq -1 \\ \ln x, & \text{якщо } n = -1 \end{cases}$$

Обчислювати F доти, поки F залишається менше 100.

Вихідні дані: a,n, z ≥ 0,4; hz=0,5.

Виведені дані: x,y,F,z, кількість обчислених значень F.



4.2. Завдання для самостійної роботи

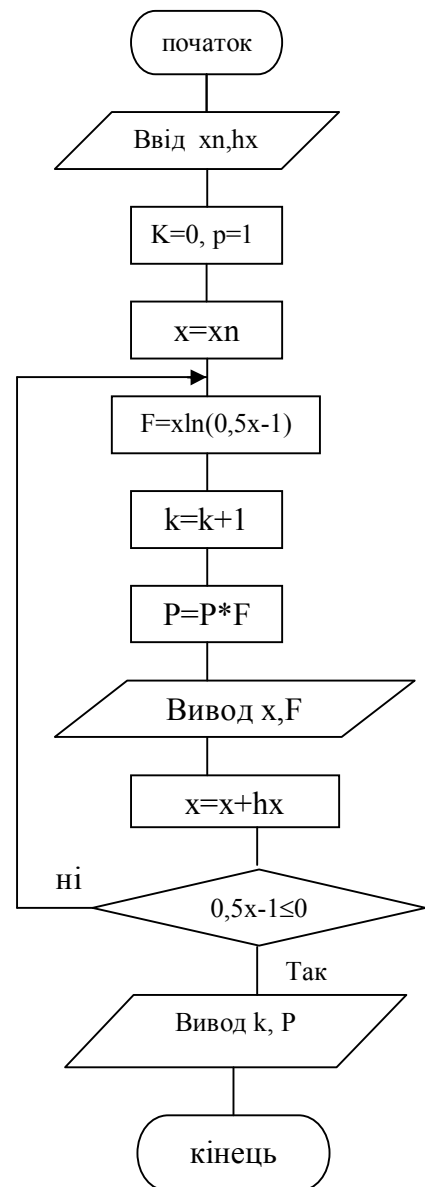
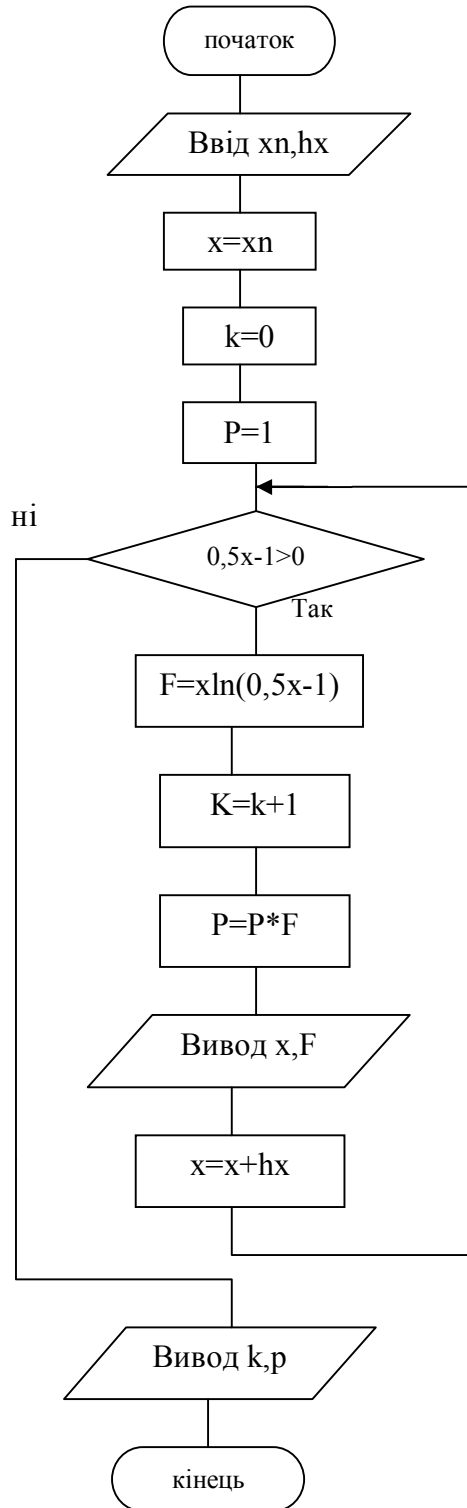
Приклад №1.

Обчислити значення виразу $F=x \cdot \ln(0.5 \cdot x - 1)$.

Обчислювати F , поки вираз під знаком логарифма >0 .

Вихідні дані: $x \leq 6$, $hx = -0,4$.

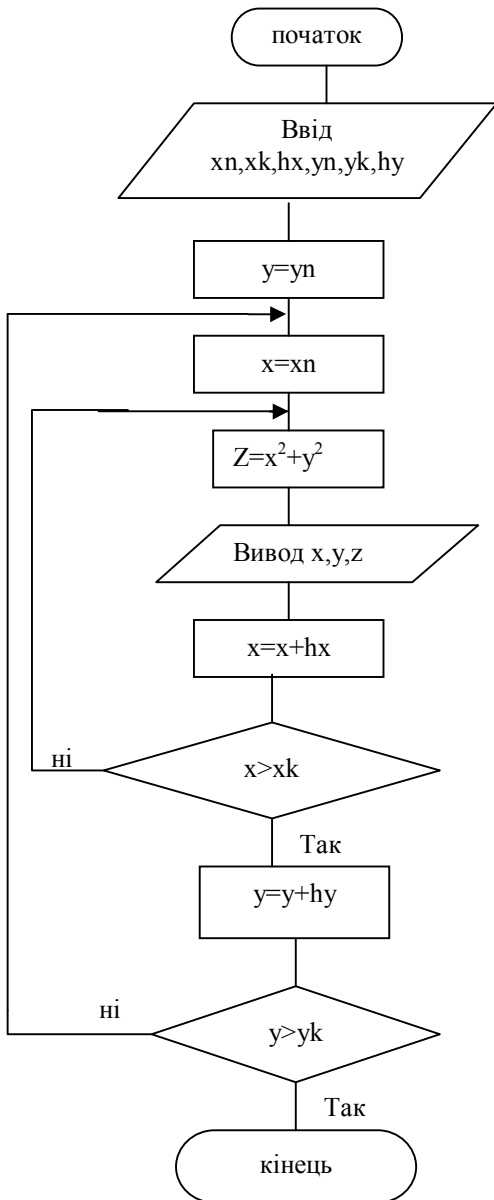
Виведені дані: F , x , P , $P = \Pi F$, кількість співмножників у P .



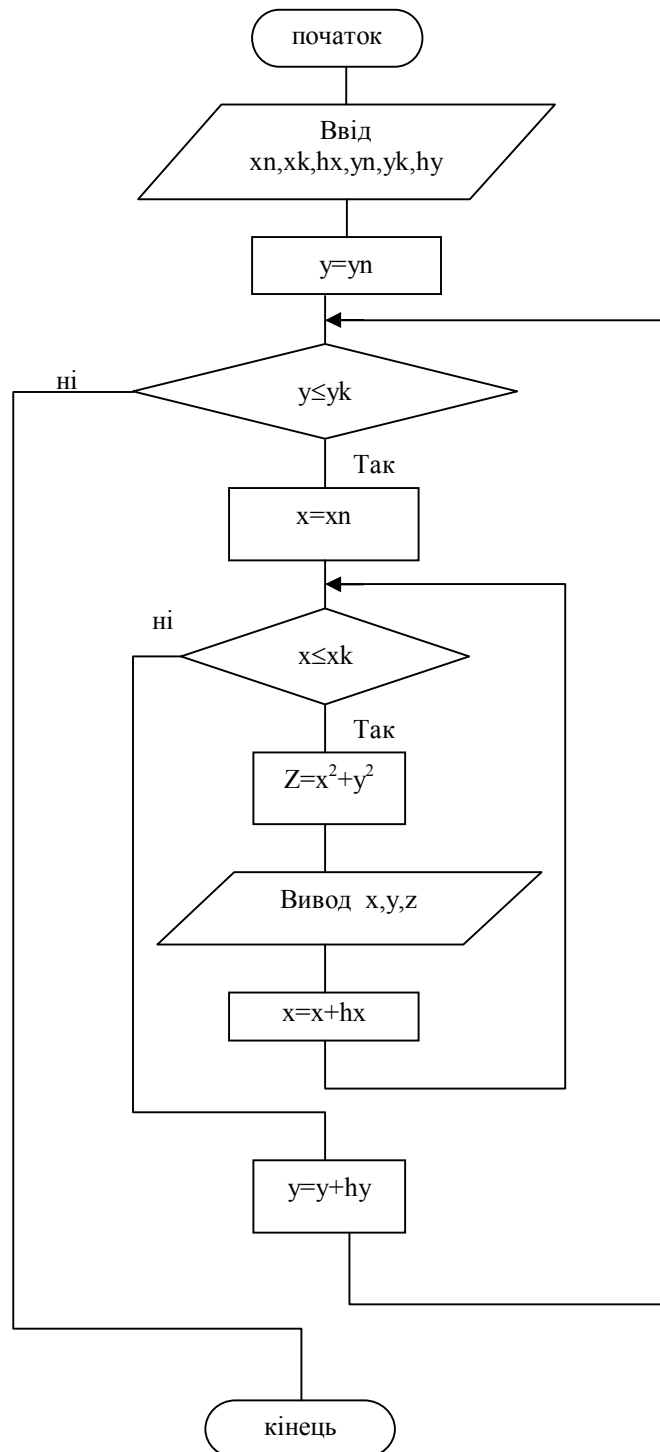
Лекція №5. ОРГАНІЗАЦІЯ ВКЛАДЕНИХ ЦИКЛІВ

Приклад. Обчислити значення виразу $Z=x^2+y^2$ при $-2 \leq x \leq 2$, $h_x=1$
 $-3 \leq y \leq 1$, $h_y=1$

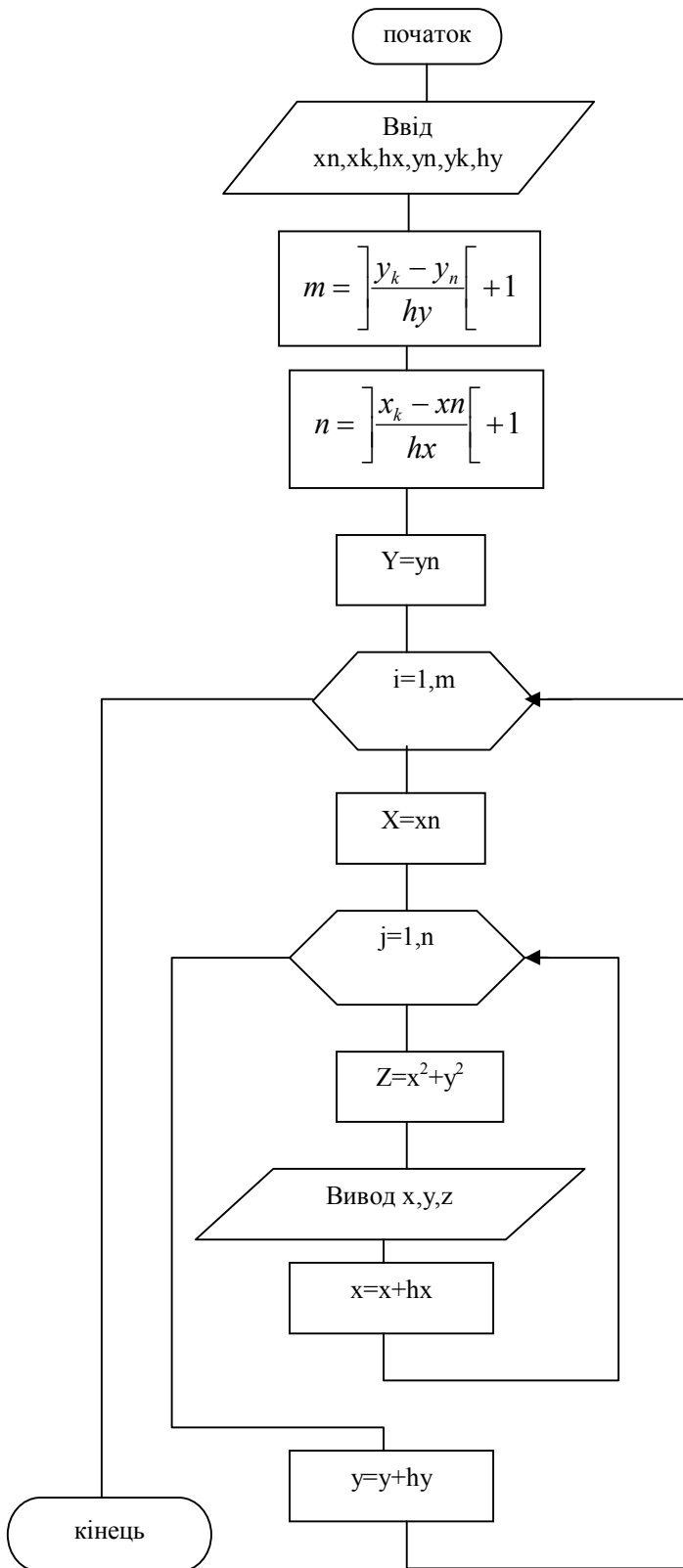
Цикл із пістумовою.



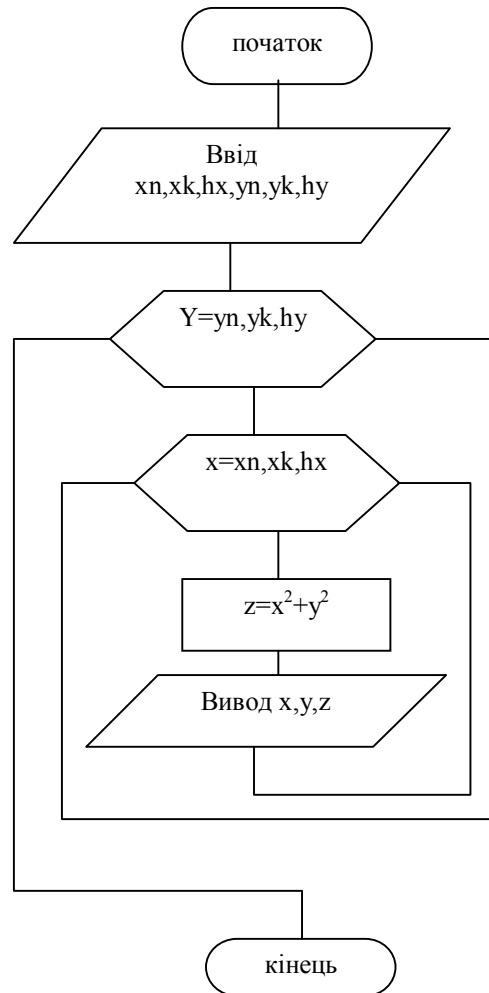
Цикл із передумовою.



Цикл «Для»

**Самостійно:**

Організувати цикл «Для» не підраховуючи кількість x і кількість y



Зауваження: При організації вкладених циклів кожний із трьох циклів можна сполучати з кожним з даних трьох циклів.

Лабораторна робота №4.

Тема: «Організація вкладених циклів»

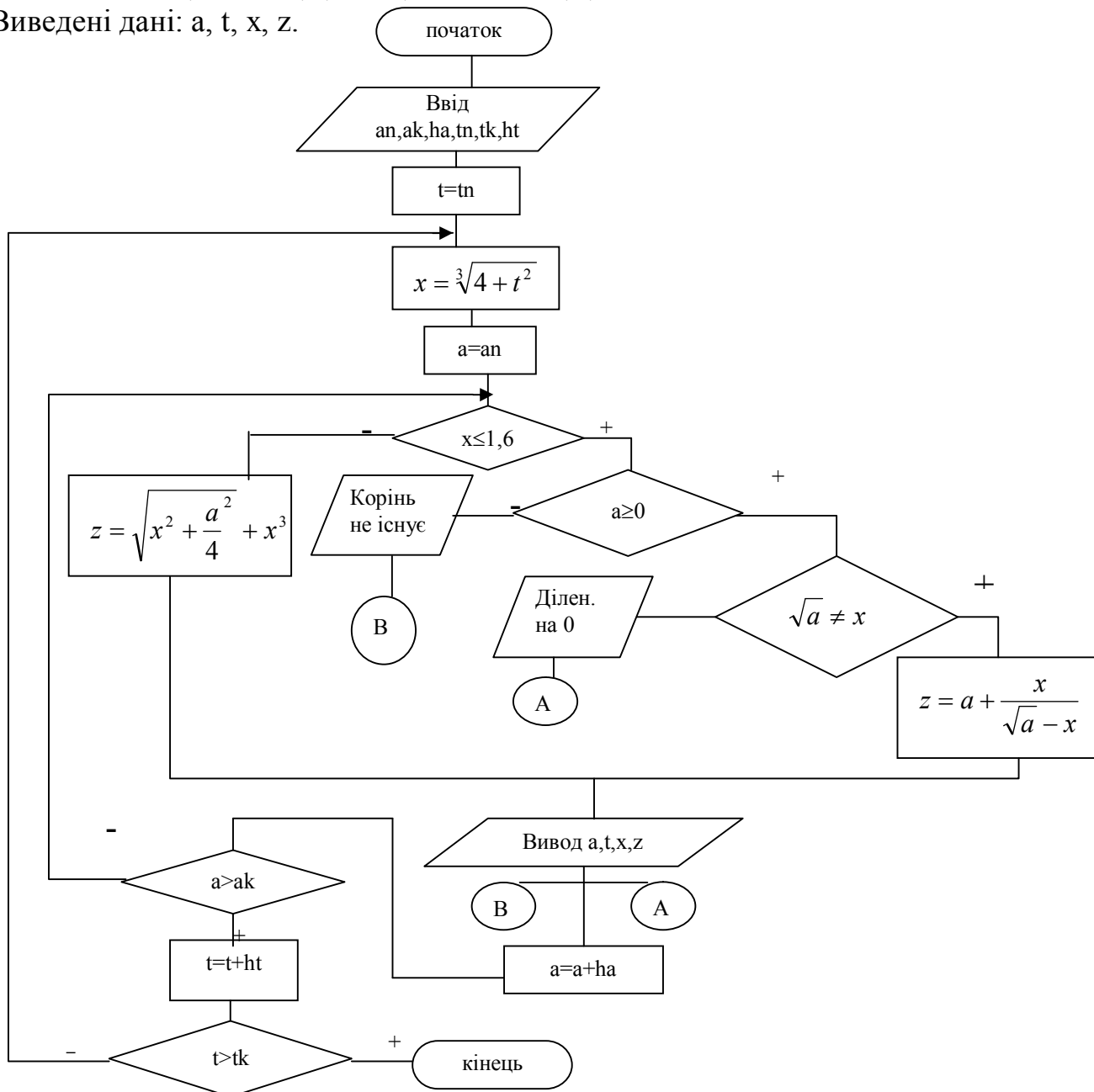
При складанні блок-схем організувати цикли з **пістумовою** або із **переду-мовою**, цикл “Для” можна організувати тільки в тому випадку, якщо змінна циклу ціле число й крок дорівнює 1.

Самостійна робота

$$Z = \begin{cases} \sqrt{x^2 + \frac{a^2}{4}} + x^3, & \text{если } x > 1,6 \\ a + \frac{x}{\sqrt{a-x}}, & \text{если } x \leq 1,6 \end{cases}, x = \sqrt[3]{4+t^2}$$

Вихідні дані: $1,5 \leq a \leq 2,5$; $h_a=0,5$ $2 \leq t \leq 4,5$; $h_t=1$

Виведені дані: a, t, x, z.



Лекція №6.

ОРГАНІЗАЦІЯ ІТЕРАЦІЙНОГО ПРОЦЕСУ

Цикл називається ітераційним, якщо при кожному наступному виконанні циклу результат обчислення наближається до шуканого із заданою точністю.

Визначення 1. Функціональним рядом називається вираз вигляду:

$$u_1(x) + u_2(x) + \dots + u_n(x) + \dots = \sum_{i=1}^{\infty} u_i(x), \text{ де } u_1(x), u_2(x), \dots \text{ є функції одного аргу-}$$

менту.

Якщо $u_i(x) = a_i x^i$ – то ряд називається статечним.

Розглянемо часткові суми:

$$S_1(x) = u_1(x), S_2(x) = u_1(x) + u_2(x), \dots, S_n(x) = u_1(x) + u_2(x) + \dots + u_n(x)$$

Визначення 2. Ряд називається збіжним (має суму), якщо послідовність його часткових сум має кінцеву межу.

$$S(x) = \lim_{n \rightarrow \infty} S_n(x) = \lim_{n \rightarrow \infty} \sum_{i=1}^n u_i(x)$$

Визначення 3. Сукупність тих значень x , для яких функціональний ряд сходиться (має суму) називається областю збіжності цього ряду, а функція $S(x)$ – його сумою. Функціональний ряд у крапці перетворюється в числовий ряд.

Надалі аргумент x будемо опускати.

$$S = u_1 + u_2 + \dots + u_n + \dots, S = S_n + u_{n+1} + u_{n+2} + \dots + \dots$$

$$R_n = u_{n+1} + u_{n+2} + \dots + \dots - \text{ залишковий член ряду.}$$

Якщо ряд сходиться, то й залишковий член ряду сходиться. $S = S_n + R_n$, $R_n = S - S_n$.

Теорема. Якщо суму ряду замінити першими n членами, то помилка, що ми отримуємо, не перевершує першого члена, що відкидає. (Якщо $S \approx S_n$, то $|R_n| < u_{n+1}$).

Якщо вибрати n так, що $u_{n+1} \leq \varepsilon$, де $\varepsilon > 0$, то $|R_n| < \varepsilon$, те говорять, що сума ряду

знайдена із заданою точністю ε .

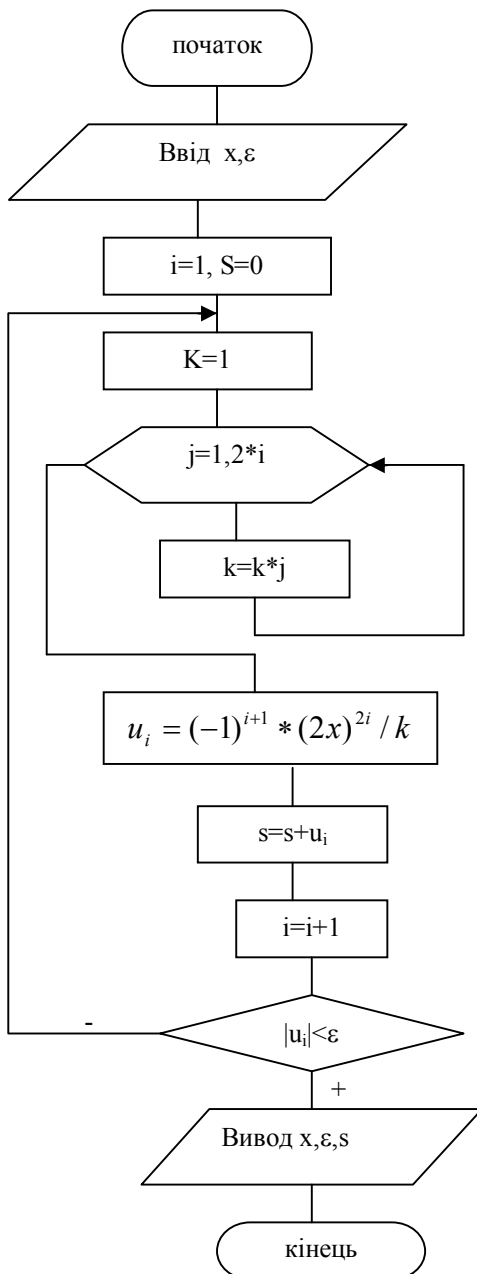
Лабораторна робота №5.

Тема: «Організація ітераційного процесу»

Приклад №1. Обчислити значення функції $y=2\sin^2x$, як знаходження суми ряду:

$$y = \frac{(2x)^2}{2!} - \frac{(2x)^4}{4!} + \frac{(2x)^6}{6!} - \dots \pm \frac{(2x)^{2i}}{(2i)!} \mp \dots \quad \text{с заданою точністю } \varepsilon=0,001; x=0,5$$

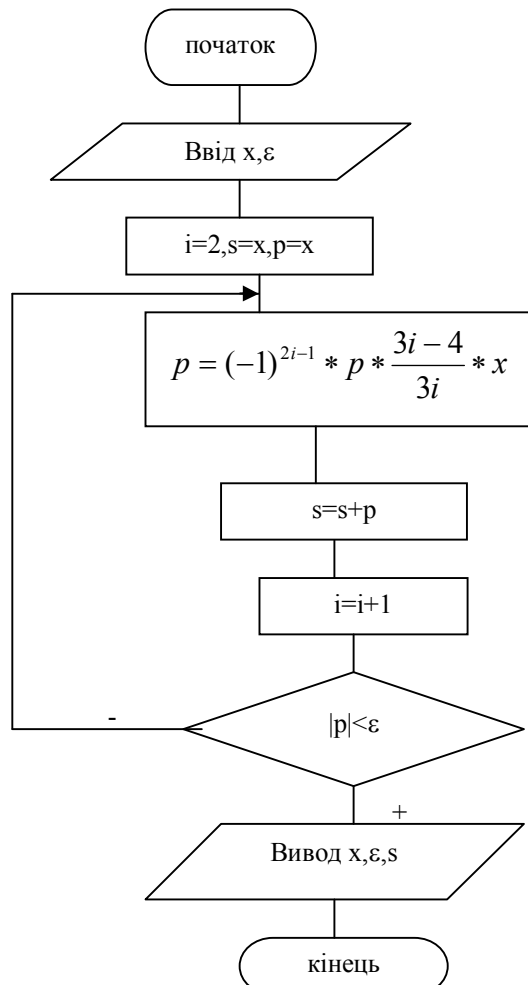
$$u_i = (-1)^{i+1} \cdot \frac{(2x)^{2i}}{(2i)!} - \text{загальний член ряду.}$$



Приклад №2. (Вар. №1)

$$y = 3\sqrt[3]{1+x} - 3$$

$$y = x - \frac{2}{6}x^2 + \frac{2 \cdot 5}{6 \cdot 9}x^3 - \dots \pm \frac{2 \cdot 5 \dots (3i-4)}{6 \cdot 9 \dots 3i}x^i \mp \dots$$



Приклад №3. (Вар №6)

$$y = 1 - \frac{1}{\sqrt[3]{1+x}} \quad y = \frac{1}{3}x - \frac{1 \cdot 4}{3 \cdot 6}x^2 + \frac{1 \cdot 4 \cdot 7}{3 \cdot 6 \cdot 9}x^3 - \dots \pm \frac{1 \cdot 4 \cdot 7 \dots (3i-2)}{3 \cdot 6 \cdot 9 \dots 3i}x^i \mp \dots$$

Початкові значення параметрів: $i=1, s=0, p=-1$

Рекурентна формула: $p = (-1) \cdot p \cdot \frac{(3i-2)}{3i}x, \quad s = s + p, \quad i = i + 1$

Приклад №4. (Вар №7) $y = \frac{48}{15}(\sqrt{(x+5)^5} - 1) - 8x - 6x^2$

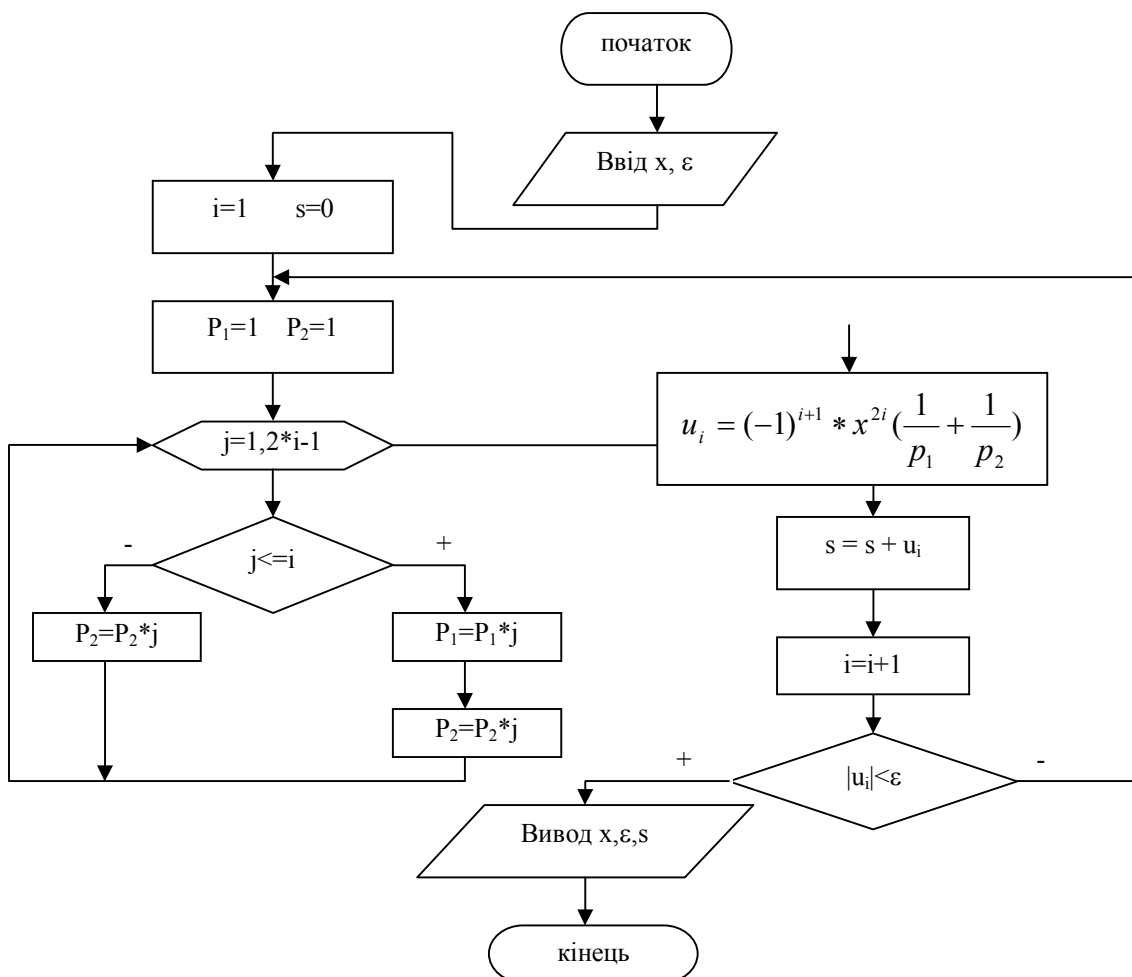
$$x^3 - \frac{1}{8}x^4 + \frac{1 \cdot 3}{8 \cdot 10}x^5 - \frac{1 \cdot 3 \cdot 5}{8 \cdot 10 \cdot 12}x^6 + \dots \mp \frac{1 \cdot 3 \cdot 5 \dots (2i-3)}{8 \cdot 10 \cdot 12 \dots 2(i+2)}x^{i+2} \pm \dots$$

Початкові значення параметрів: $i = 2, \quad s = x^3, \quad p = x^3,$

Рекурентна формула: $p = (-1) * p * \frac{2i-3}{2(i+2)} * x, \quad s = s + p, \quad i = i + 1$

Приклад №5. (Вар №2). $y = x \sin x - e^{-x^2} + 1$

$$y = x^2 \left(\frac{1}{1!} + \frac{1}{1!} \right) - x^4 \left(\frac{1}{2!} + \frac{1}{3!} \right) + \dots \pm x^{2i} \left(\frac{1}{i!} + \frac{1}{(2i-1)!} \right) \mp \dots$$



Приклад №6. (Вар. №3)

$$y = \sin x - \cos x + 1$$

$$y = \frac{x(2+x)}{2!} - \frac{x^3(4+x)}{4!} + \frac{x^5(6+x)}{6!} - \dots \pm \frac{x^{2i-1}(2i+x)}{(2i)!} \mp \dots$$

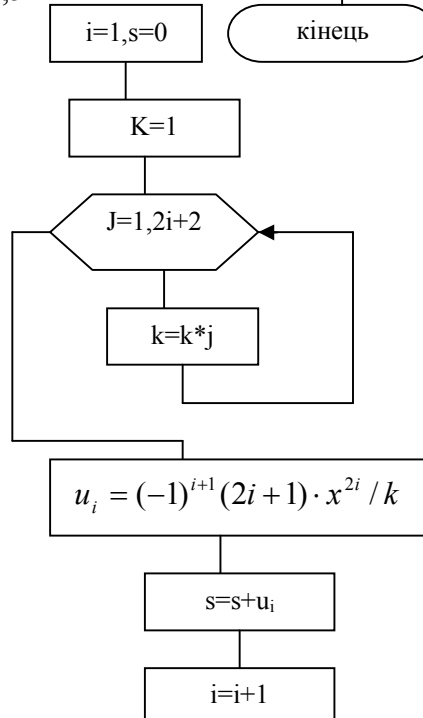
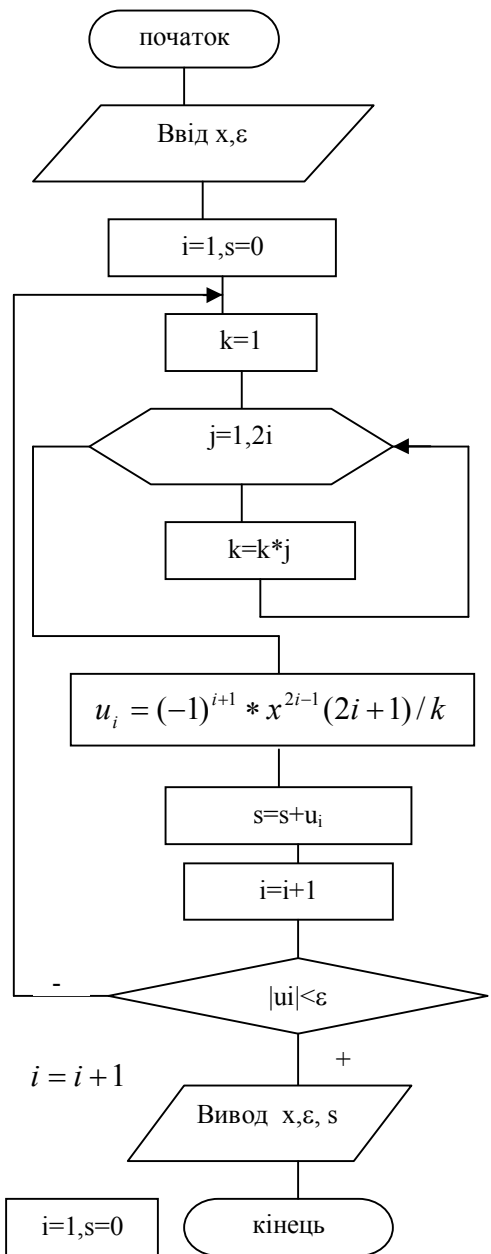
Приклад №7. (Вар. №4) $y = 1 - \frac{1}{\sqrt[4]{1+x}}$

$$y = \frac{1}{4}x - \frac{1 \cdot 5}{4 \cdot 8}x^2 + \frac{1 \cdot 5 \cdot 9}{4 \cdot 8 \cdot 12}x^3 - \dots \pm \frac{1 \cdot 5 \cdot 9 \dots (4i-3)}{4 \cdot 8 \cdot 12 \dots 4i}x^i \mp \dots$$

$$i = 1, \quad s = 0, \quad p = -1, \quad p = (-1) * p * \left(\frac{4i-3}{4i}\right) * x, \quad s = s + p, \quad i = i + 1$$

Приклад №8. (Вар. №5) $y = \frac{1 - \cos x + x \sin x}{x^2} + 0,5$

$$y = \frac{3x^2}{4!} - \frac{5x^4}{6!} + \frac{7x^6}{8!} - \frac{9x^8}{10!} + \dots \pm \frac{(2i+1) \cdot x^{2i}}{(2i+2)!} \mp \dots$$



Приклад №9. (Вар. №8) $y = (1 + x^2) \operatorname{arctg} x - x$

$$\frac{2x^3}{4 \cdot 1^2 - 1} - \frac{2x^5}{4 \cdot 2^2 - 1} + \frac{2x^7}{4 \cdot 3^2 - 1} - \dots \pm \frac{2 \cdot x^{2i+1}}{4i^2 - 1} \mp \dots$$

Початкові значення параметрів: $i = 1, \quad s = 0$

$u_i = (-1)^{i+1} \cdot 2 \cdot x^{2i+1} / (4i^2 - 1), \quad s = s + u_i, \quad i = i + 1$ - загальний член ряду.

Лекція №7. ПОНЯТТЯ ІНДЕКСОВАНОЇ ЗМІННОЇ.

МАСИВИ

Масив - це послідовний набір даних одного типу, що має одне ім'я.

$$A = \{a_1, a_2, \dots, a_N\}.$$

Масив - це структурований тип даних, що складається з фіксованого числа елементів.

Всі елементи масиву пронумеровані. Номер елемента масиву називається його індексом. Доступ до кожного елемента масиву здійснюється за допомогою індексів. Наприклад, $A(3), B(3,2)$. Таким чином, масив - це індексована змінна.

Кількість елементів у масиві називається його розмірністю. Наприклад:

$A(N)$ - заданий одномірний масив розмірності N .

$A(M,N)$ - заданий двовимірний масив розмірності $M \times N$ - що складається з M рядків і N стовпців.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{array}{l} A(2,4) \text{- це елемент масиву, який стоїть на} \\ \text{перетинанні другого рядка й четвертого} \\ \text{стовпця} \end{array}$$

На практиці одномірний масив - це вектор, двовимірний масив - це матриця, що складається з визначеної кількості рядків і стовпців.

Обробка масивів звичайно полягає в послідовному переборі їхніх елементів і виконанні над кожним елементів тих самих дій, тобто для перебору елементів масиву необхідно організувати цикл.

Ввід і вивід масивів.

Ввід і вивід здійснюється тільки по елементне (рис. 1 і 2).

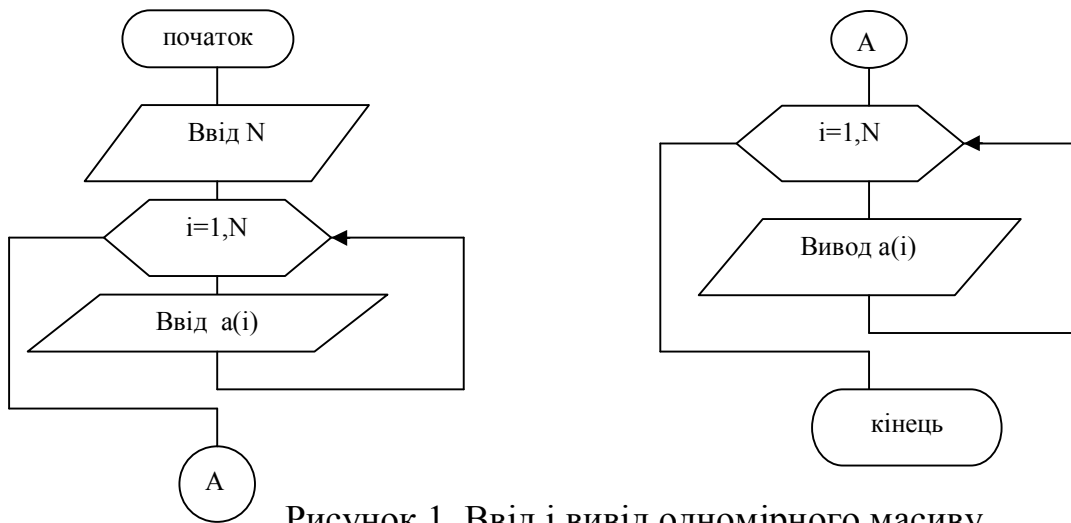


Рисунок 1. Ввід і вивід одномірного масиву

Для обробки двовимірних масивів необхідно організувати два вкладених цикли.

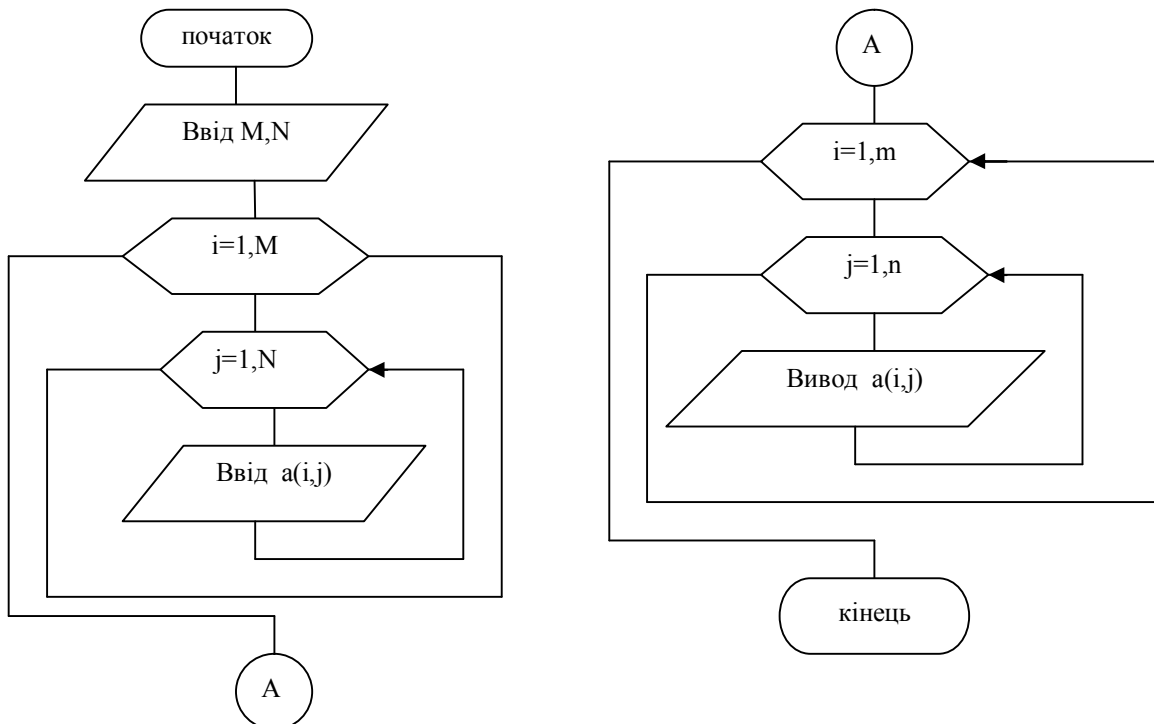


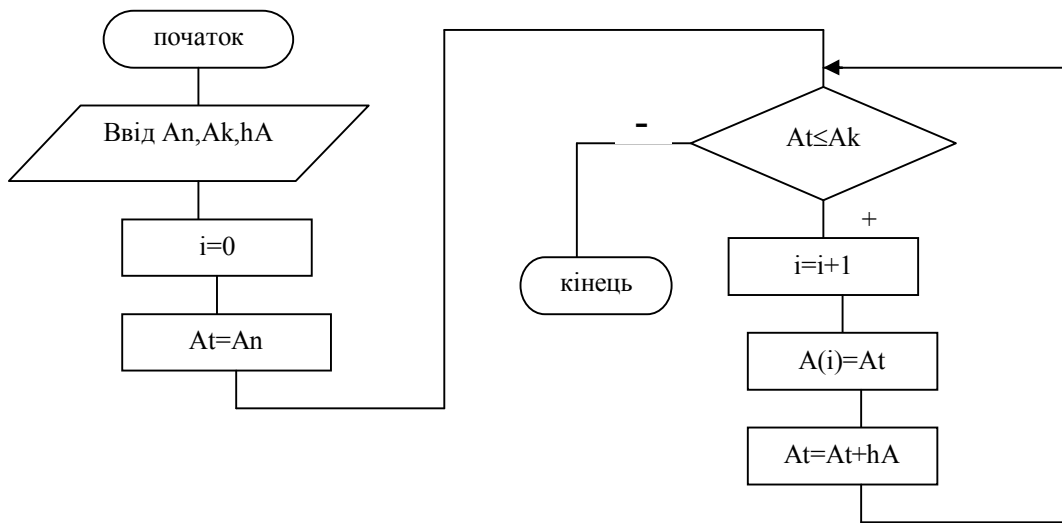
Рисунок 2. Ввід і вивід двовимірного масиву

Якщо організувати зовнішній цикл по j (номер стовпця), а внутрішній цикл по i (номер рядка) то ввід і вивід масиву буде здійснюватися по стовпцях.

Якщо елементи масиву задані рядом чисел, наприклад $A_n \leq A(i) \leq A_k$ і зада-

ний крок hA , то заповнення масиву можна здійснювати двома способами:

1. Організувати цикл із пістумовою або із передумовою, параметром якого є значення A на заданому інтервалі.

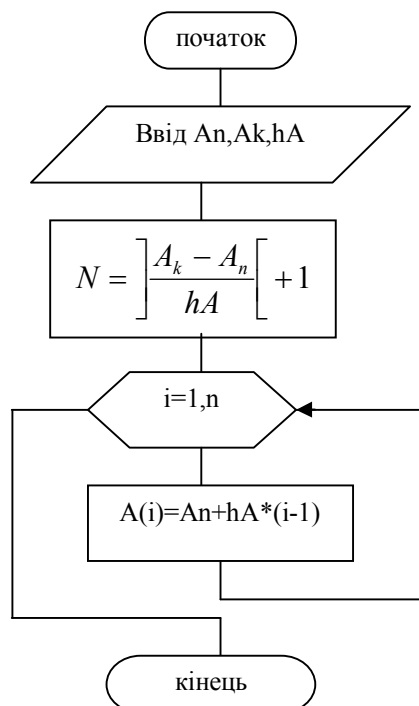


2. Організувати цикл “Для”, параметром якого є номер елемента масиву i . До початку циклу необхідно обчислити N - розмірність масиву по формулі:

$$N = \left\lceil \frac{Ak - An}{hA} \right\rceil + 1$$

Усередині циклу кожен елемент масиву визначити по формулі:

$$A_i = A_n + hA \cdot (i-1).$$

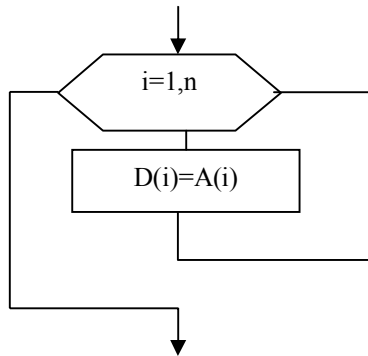


Копіювання масивів.

Є два способи копіювання масивів:

1) $\langle \text{ім'я масиву 1} \rangle = \langle \text{ім'я масиву 2} \rangle$. Наприклад $D=A$, D і A повинні належати до одного типу.

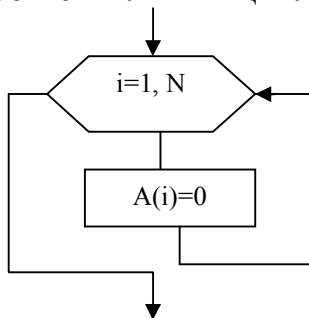
2) Копіювання поелементно в циклі.



Для двовимірних масивів копіювання елементів виконується аналогічно.

Ініціалізація масивів.

Ініціалізація означає присвоєння елементам масиву того самого значення й виконується тільки в циклі поелементно.

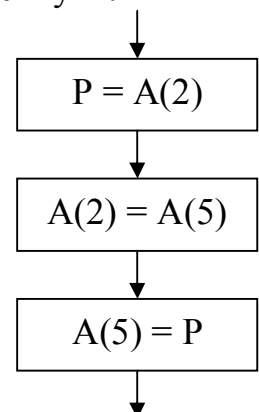


Для двовимірних масивів виконується аналогічно

Перестановка значень елементів масиву.

Перестановка елементів масиву здійснюється за допомогою додаткової змінної того ж типу, що й тип елементів масиву.

Наприклад. Поміняти місцями другий і п'ятий елемент масиву A .



Лабораторна робота №6

Тема: «Обробка одномірних масивів»

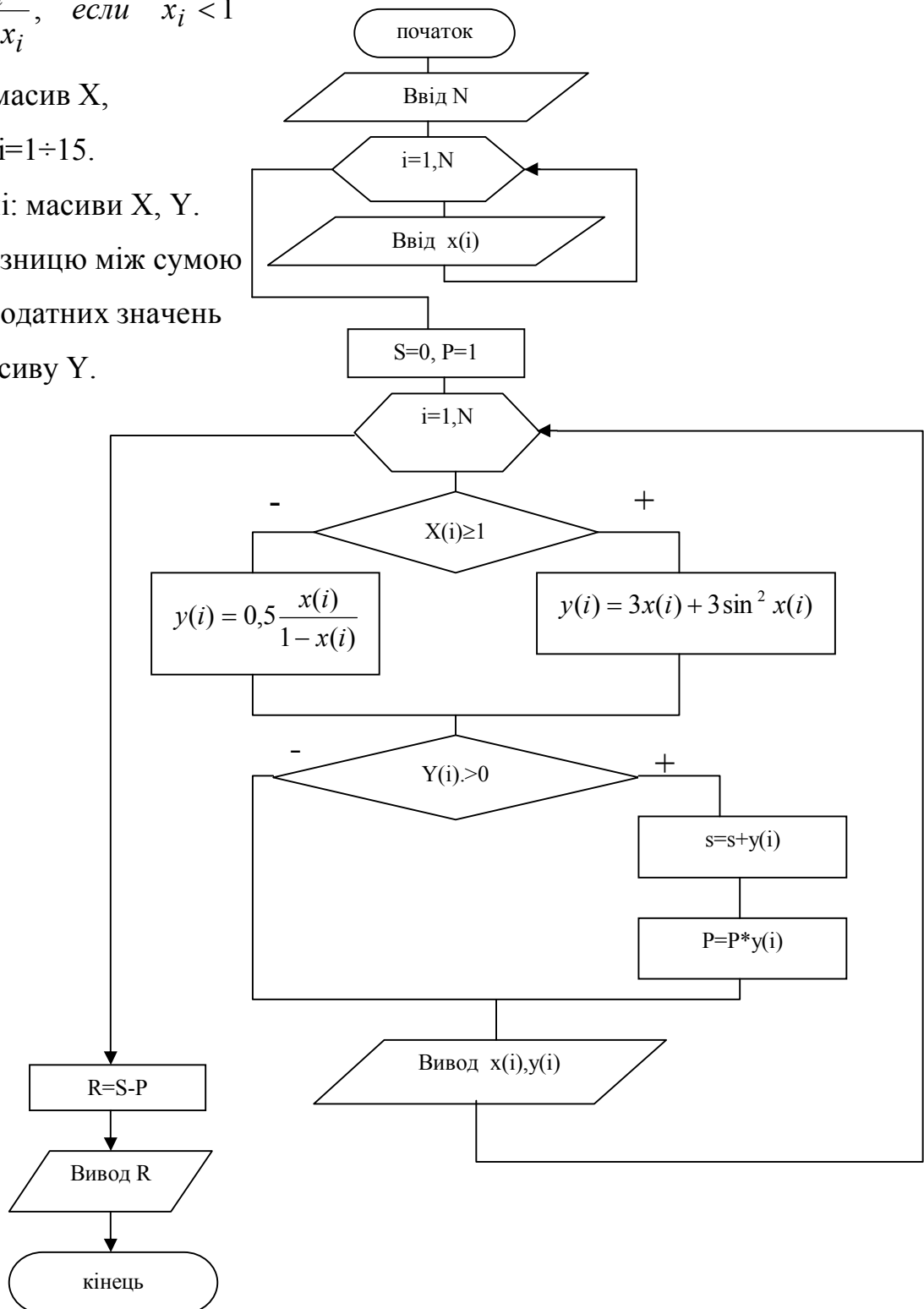
Виконати формування масиву Y відповідно до заданого виразу:

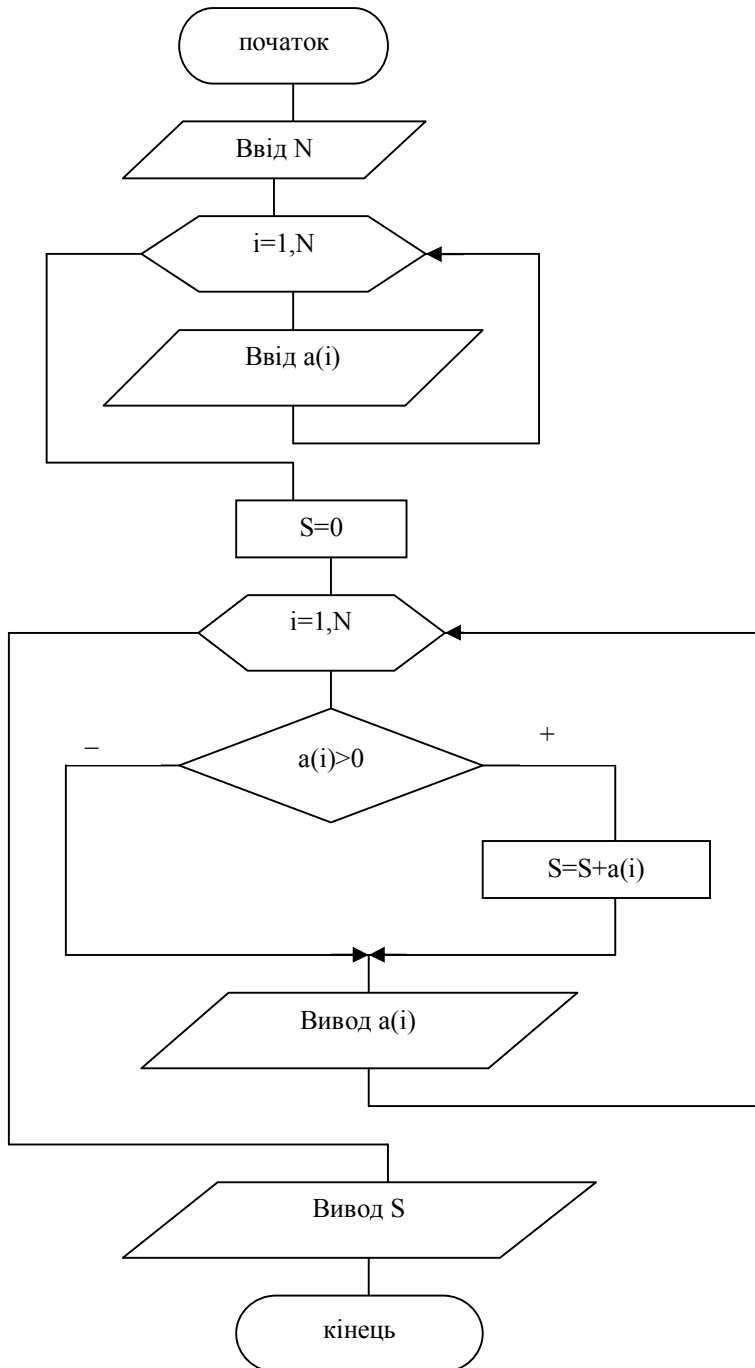
$$y_i = \begin{cases} 3x_i + 3 \sin^2 x_i, & \text{если } x_i \geq 1 \\ 0,5 \frac{x_i}{1-x_i}, & \text{если } x_i < 1 \end{cases}$$

Вхідні дані: масив X,
розмірністю $i=1 \div 15$.

Виведені дані: масиви X, Y.

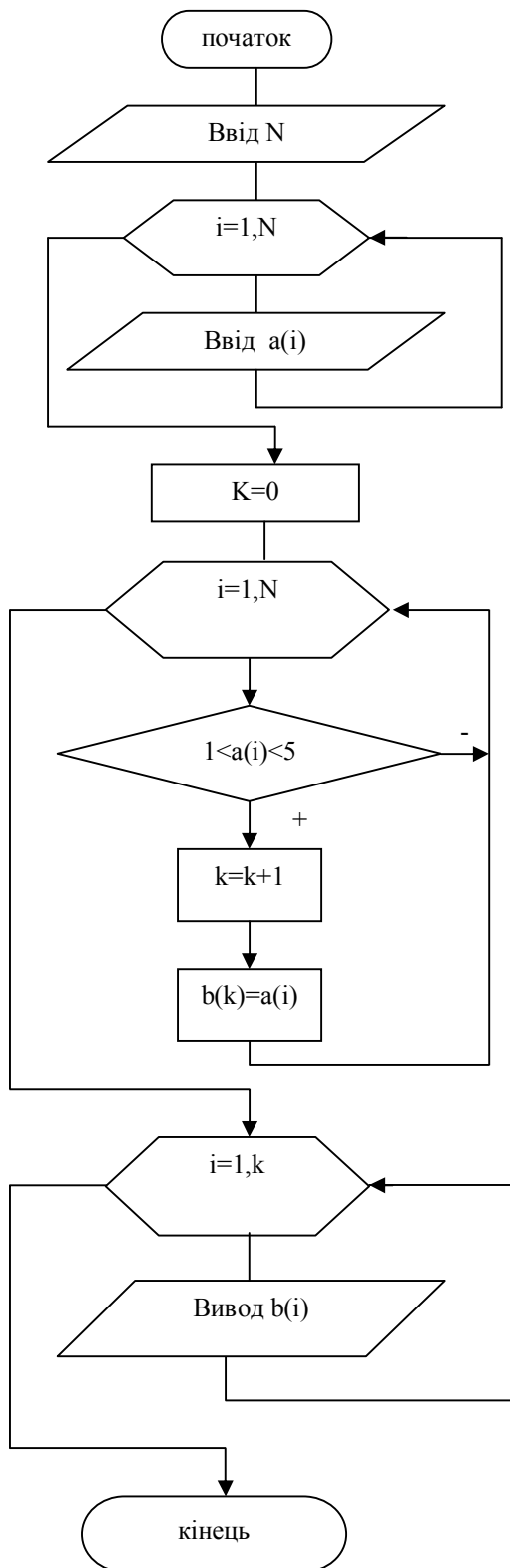
Визначити різницю між сумою
й добутком додатних значень
елементів масиву Y.



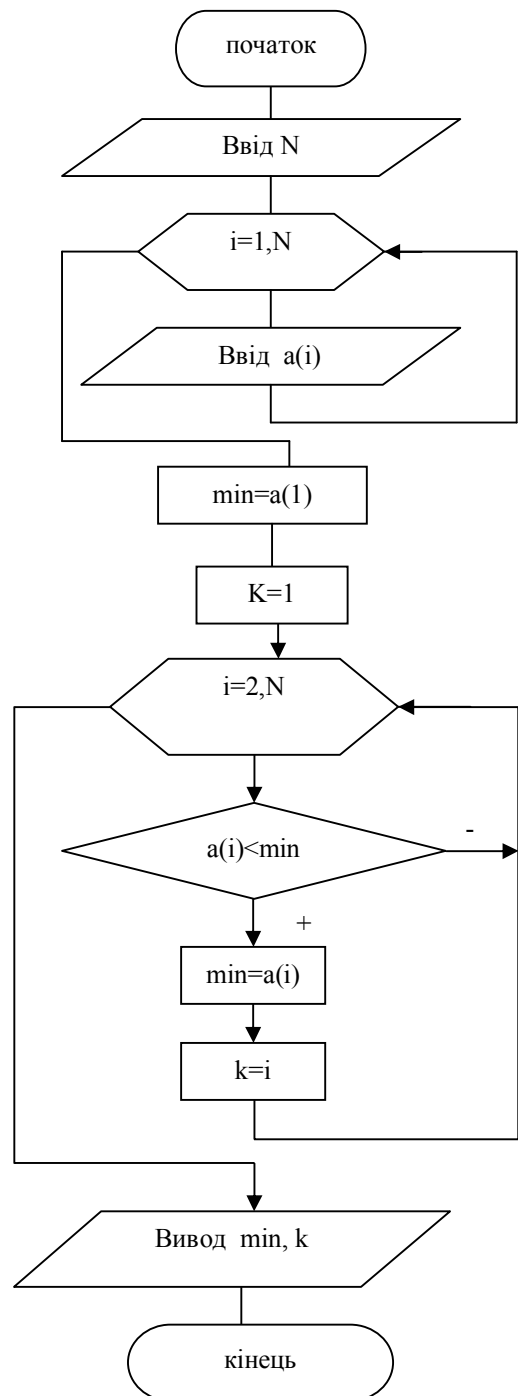
*Приклади обробки одномірних масивів.***Приклад №1.**(Самостійно).Знайти суму додатних елементів масиву $A(N)$.

Лекція №8. ПРИКЛАДИ ОБРОБКИ ОДНОМІРНИХ МАСИВІВ

Приклад №1. У масив В помістити елементи масиву А(N), значення яких більше 1 і менше 5, тобто $1 < a(i) < 5$.

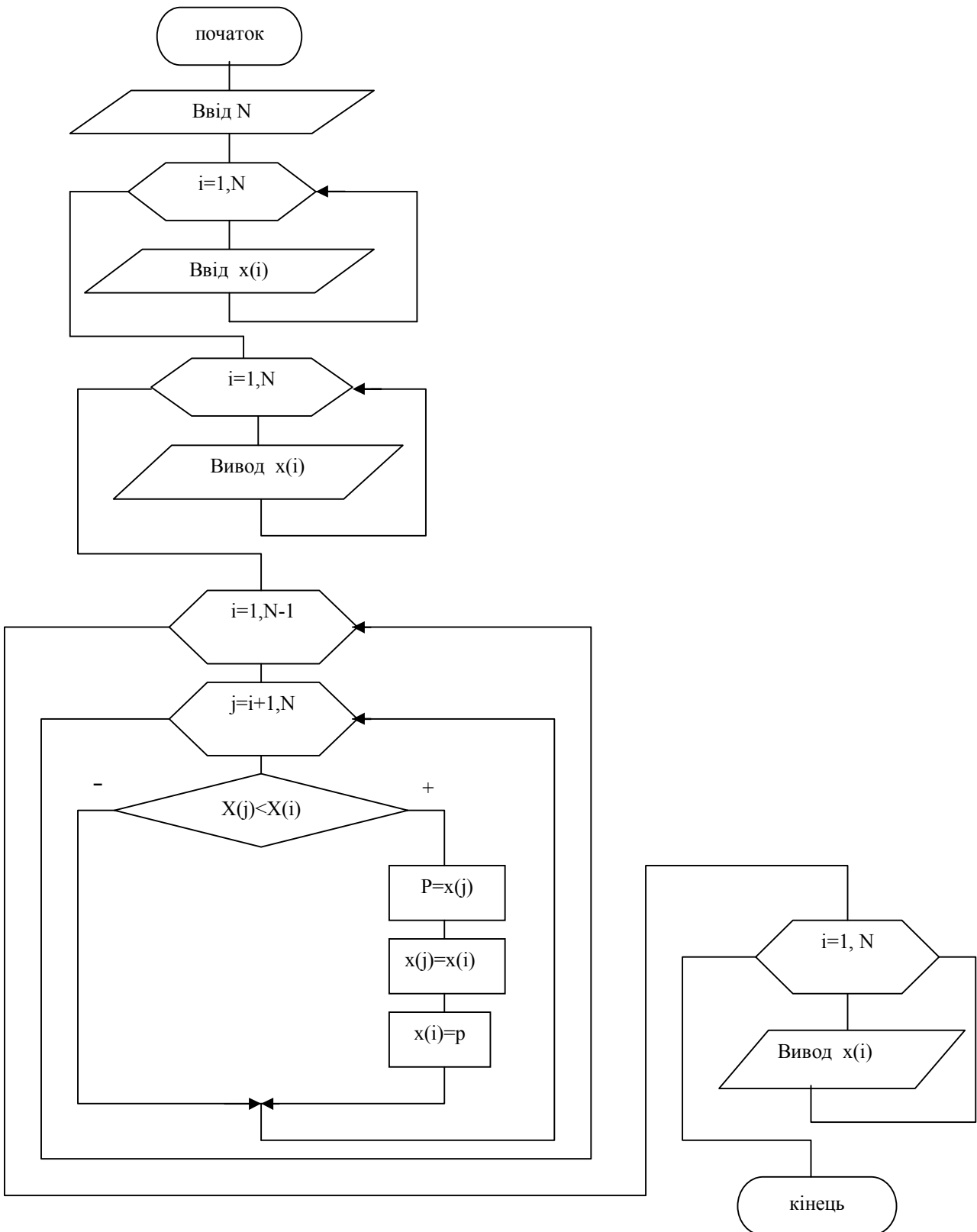


Приклад №2. Знайти мінімальний елемент масиву А(N) і визначити його номер.

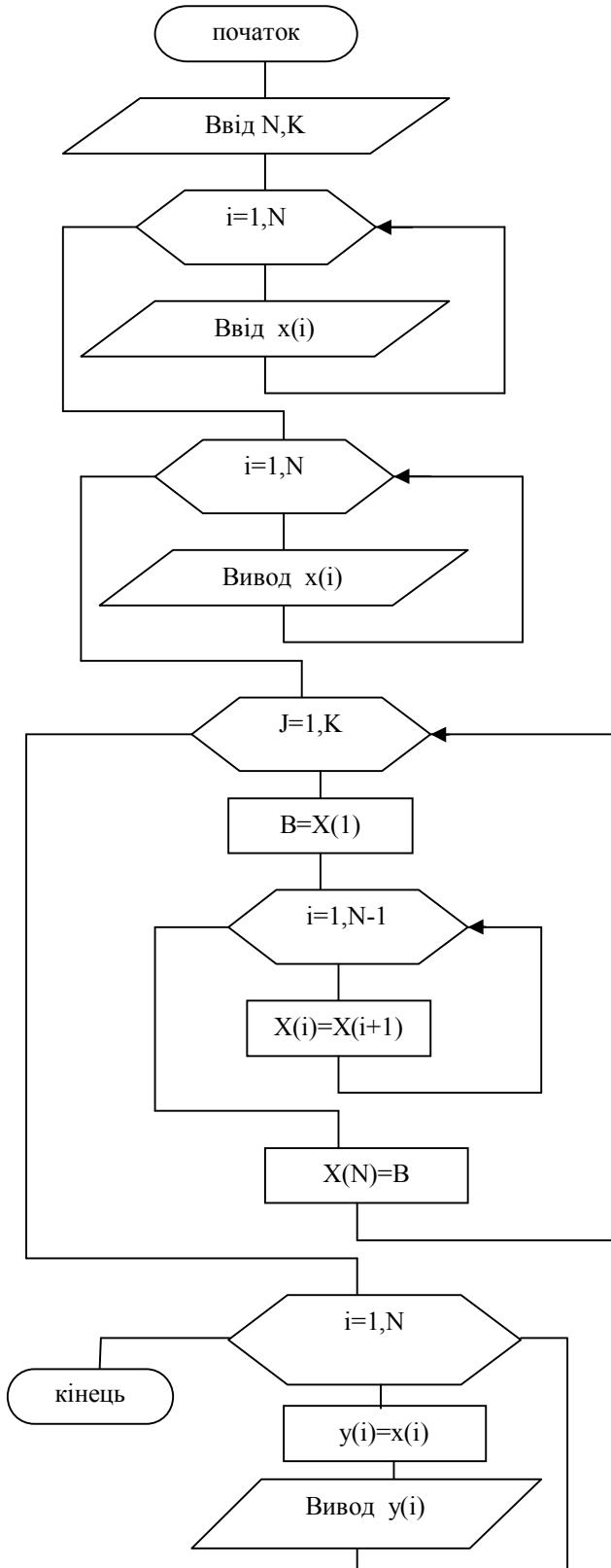


Приклад №3.

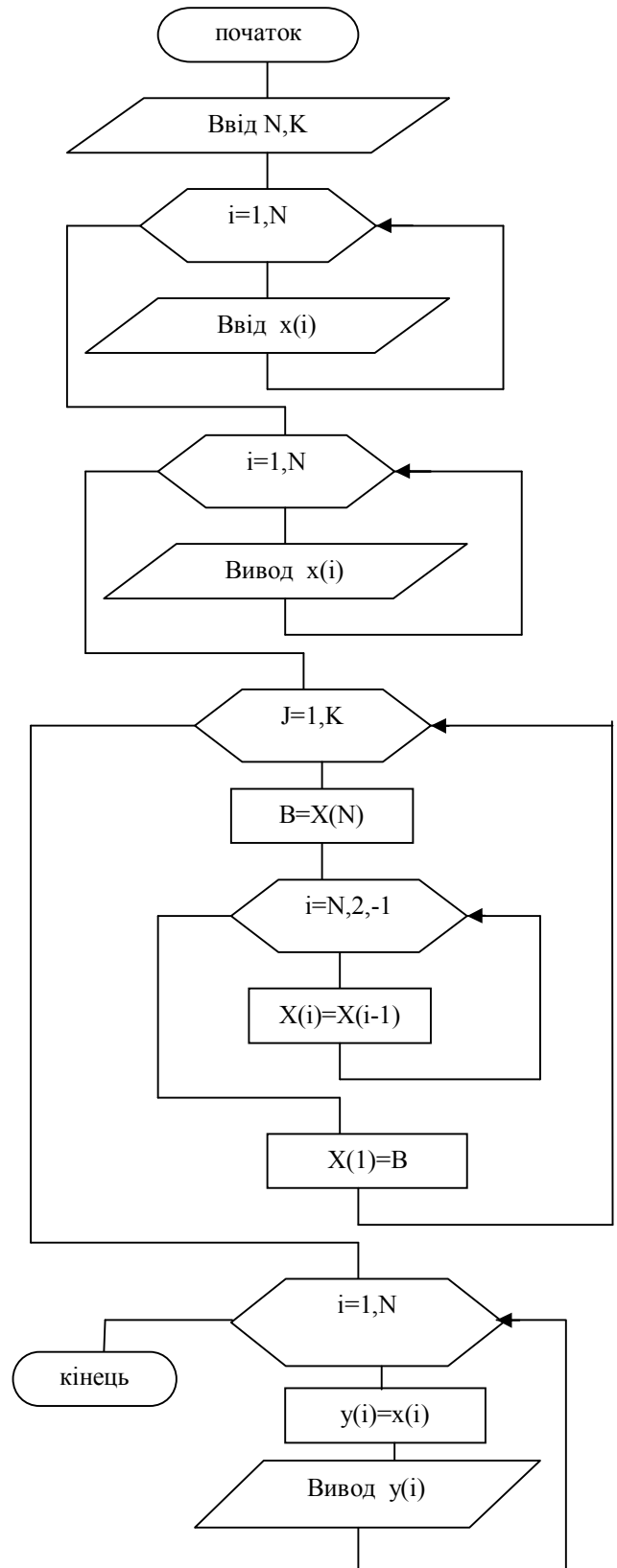
Елементи масиву $X(N)$ упорядкувати по зростанню.



Приклад №4. Записати елементи масиву $X(N)$ у масив $Y(N)$, зрушивши елементи масиву X уліво на k позицій.

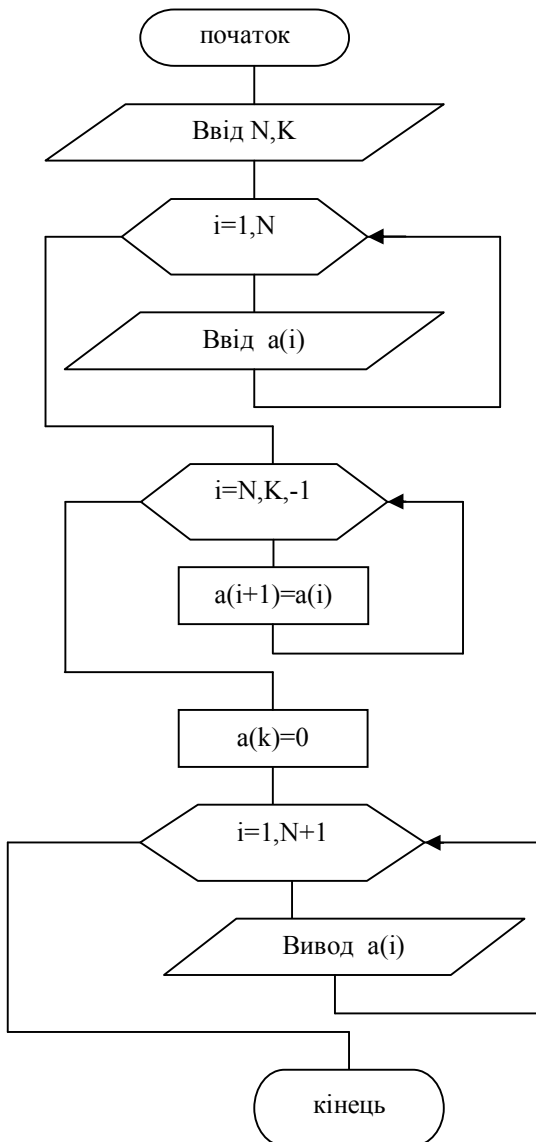


Приклад №5. Записати елементи масиву $X(N)$ у масив $Y(N)$, зрушивши елементи масиву X вправо на k позицій.



Завдання на самостійну роботу.

Приклад №1. В одномірний масив $A(N)$ вставити k -й елемент рівний 0.



Додавання одного елемента збільшує розмірність масиву, тобто $N+1$.

Елементи вхідного масиву розташовані: $a_1, a_2, a_3, \dots, a_N$. Перш ніж вставити новий елемент, ми повинні звільнити для нього місце. Для цього всі елементи масиву, починаючи з a_k , необхідно перемістити на одну позицію вправо, причому зсув треба починати з останнього елемента:

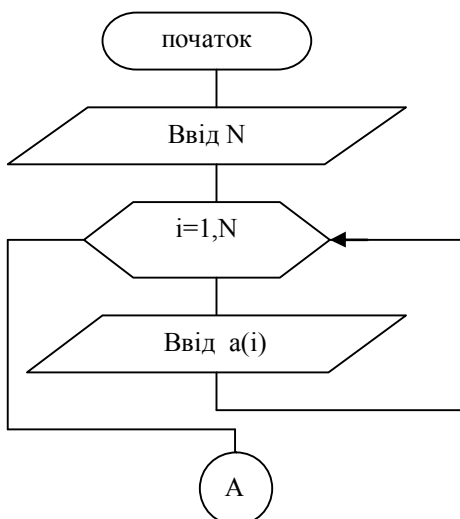
$$a(N) \xrightarrow{\text{перемістити}} a(N+1)$$

$$a(N-1) \longrightarrow a(N)$$

... ..

$$a(k) \longrightarrow a(k+1)$$

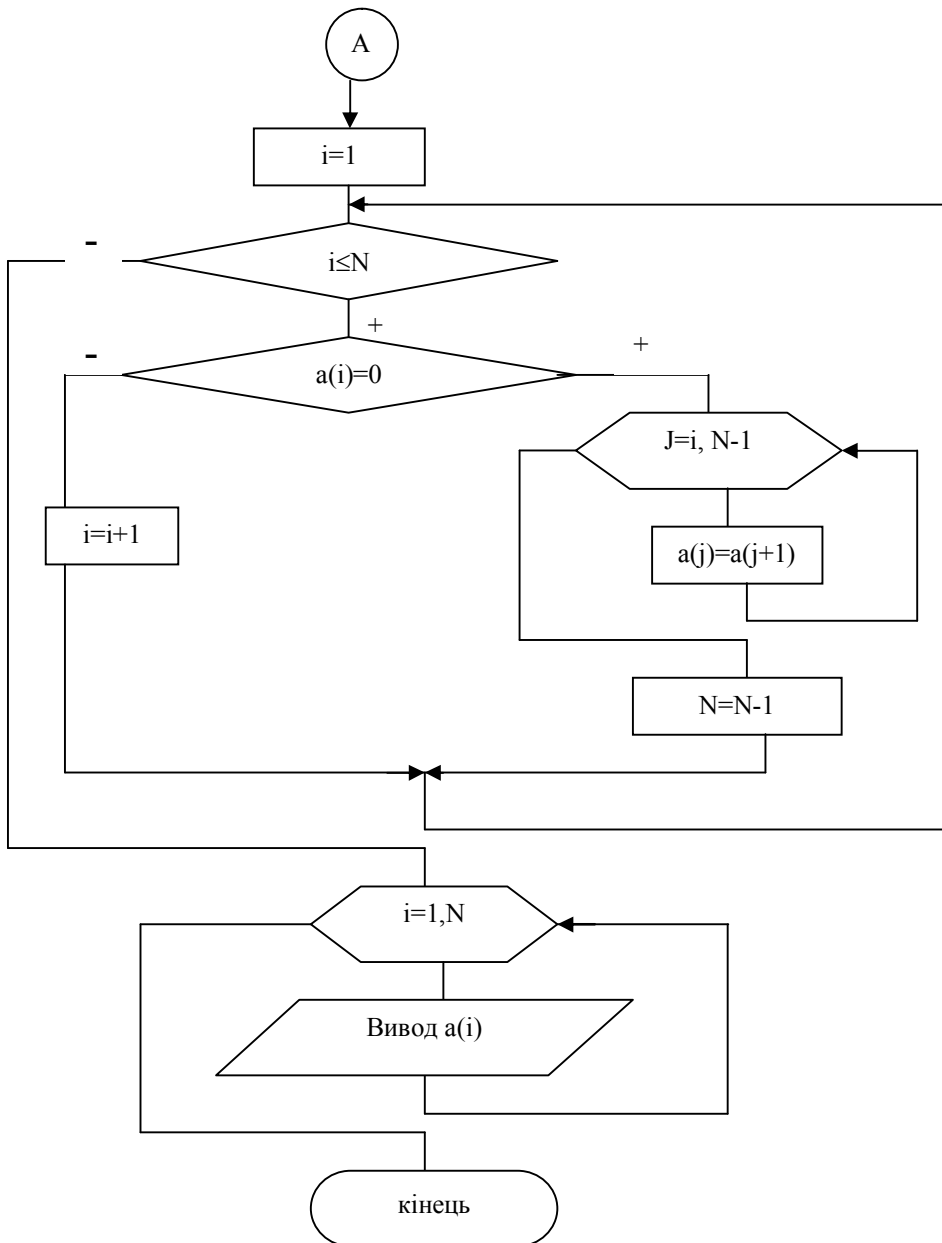
Після цих переміщень у масиві елементи $a(k)$ і $a(k+1)$ однакові. Тепер k -му елементу можемо присвоїти нове значення, за умовою 0.



Фізично ми не можемо видалити елемент, не змінюючи положення інших. Тому якщо необхідно видалити визначений елемент, наприклад елемент із номером 5, ми повинні всі елементи, які починаються із шостого й до останнього змістити на одну позицію вліво. При цьому a_5 прикмет значення a_6

$$a_6 \longrightarrow a_5$$

$$a_7 \longrightarrow a_6 \text{ и т. д.}$$



Останній i передостанній елемент виявляються однаковими. Скоротивши розмірність на 1, ми виключимо з розгляду останній елемент i в такий спосіб зробимо видалення.

Якщо видаляємо кілька елементів, то організуємо складний цикл. У тілі внутрішнього циклу виконується поелементне присвоювання наступного елемента попередньому (зсув вліво).

По закінченню внутрішнього циклу необхідно зменшити розмірність на 1 і перейти до зовнішнього циклу без зміни параметра зовнішнього циклу, тому що на місці вилученого може виявитися нульовий елемент.

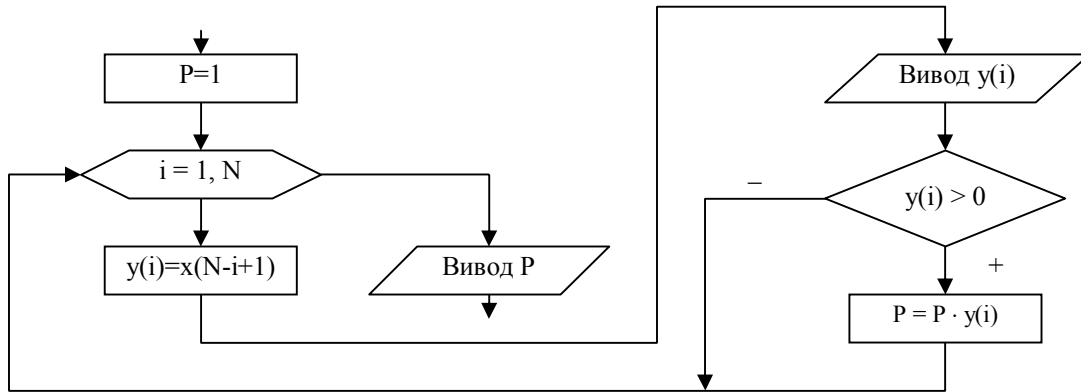
Для організації зовнішнього циклу блок модифікації використовувати не можна, тому що змінюється розмірність масиву.

Лабораторна робота №7.

Тема: «Обробка одномірних масивів з перестановкою елементів»

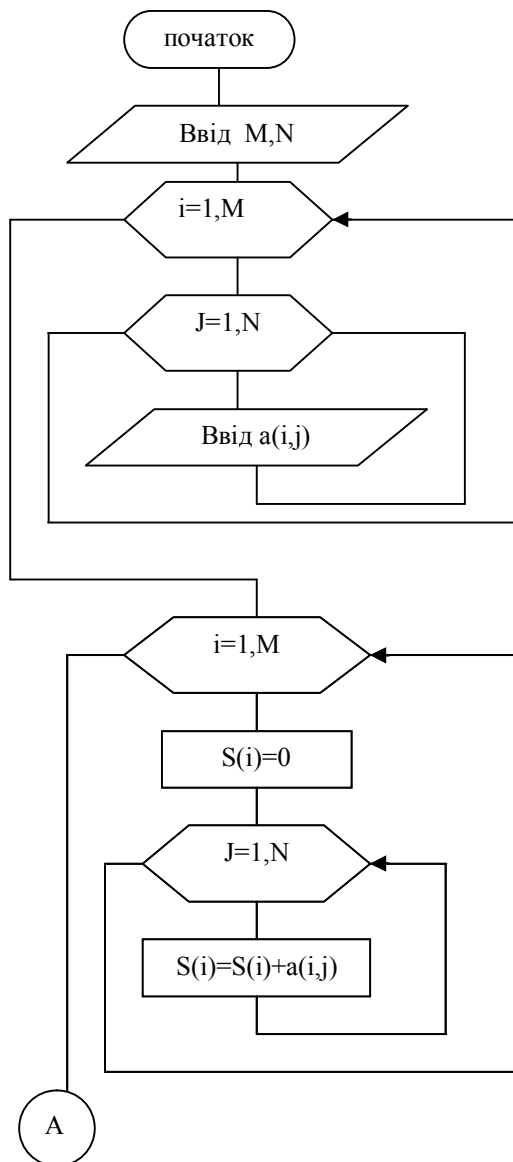
Приклад №1. Записати елементи масиву X у зворотному порядку в масив Y , об-

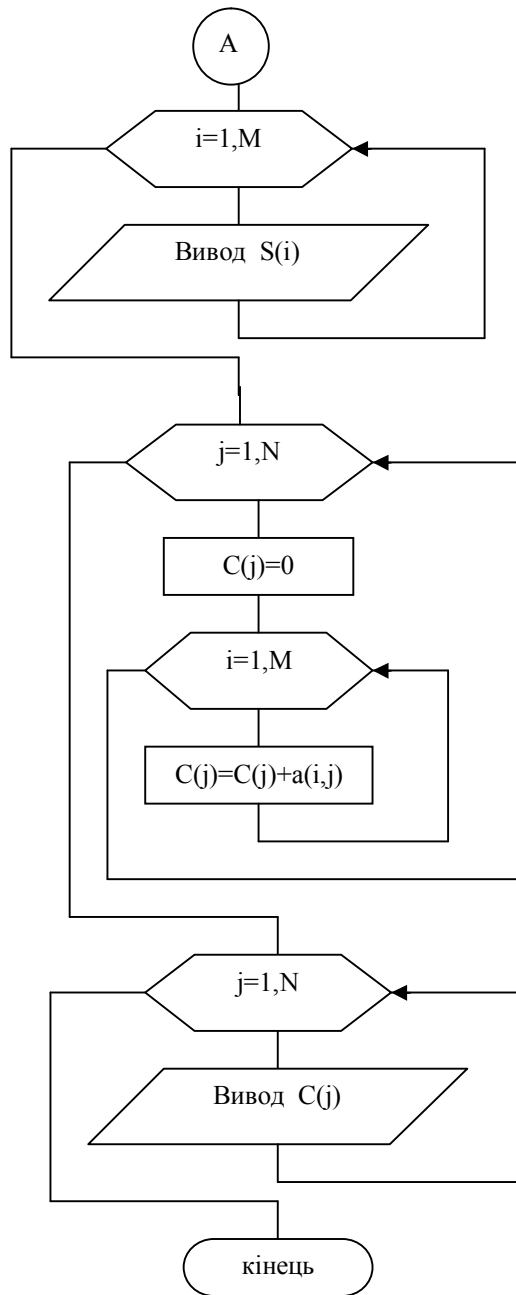
числити добуток додатних елементів масиву Y.



Лекція №9. ОБРОБКА ДВОВИМІРНИХ МАСИВІВ

Приклад №1. Знайти суму елементів кожного рядка й кожного стовпця матриці $A(M,N)$.

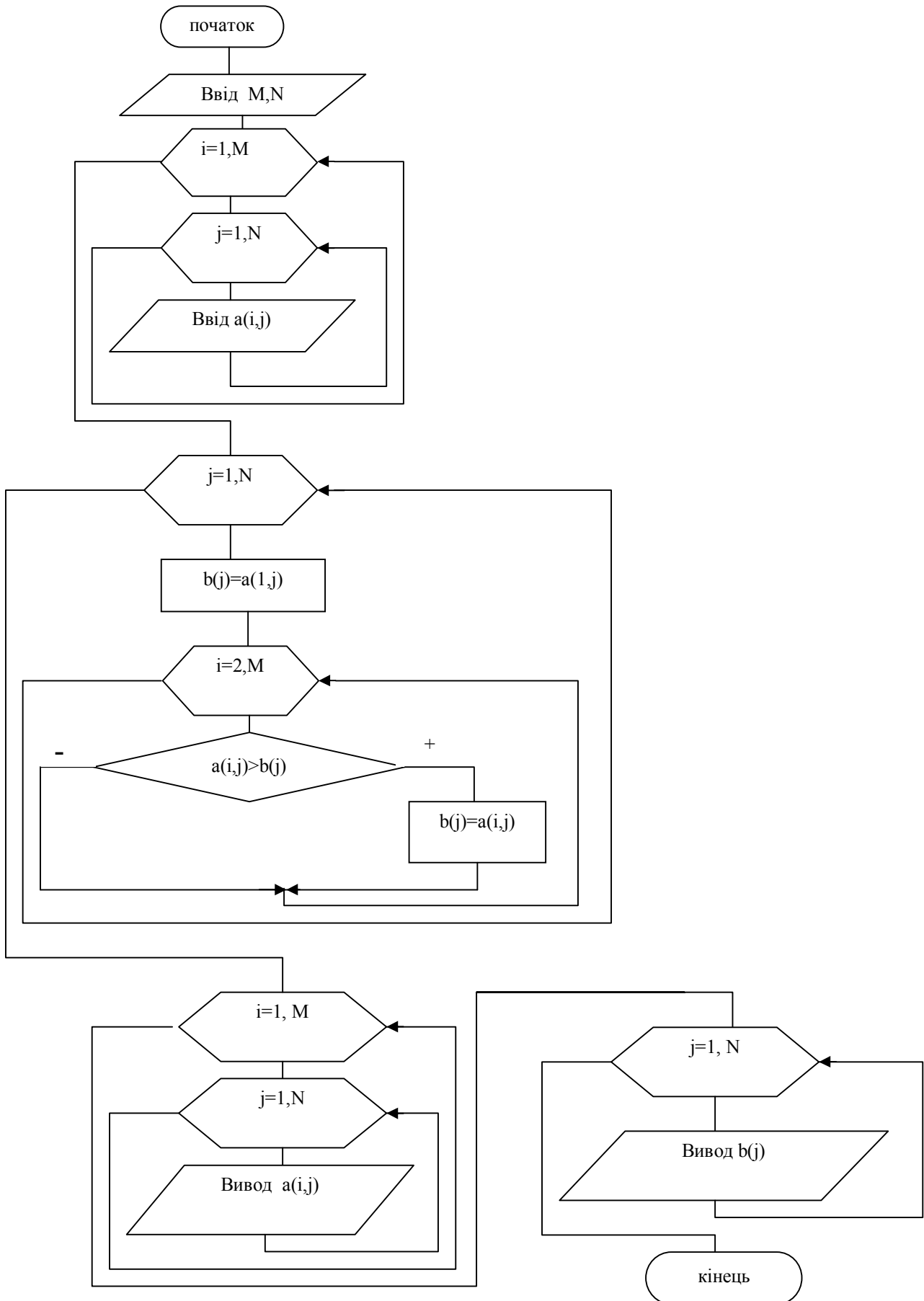




$$A = \begin{pmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \dots a_{mn} \end{pmatrix} \begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_m \end{matrix}$$

$$C_1 \quad C_2 \dots C_n$$

Приклад №2. Знайти максимальний елемент у кожному стовпці матриці $A(M,N)$.



Приклад №3. Знайти суму елементів квадратної матриці $A(N,N)$, що лежать вище головної діагоналі.

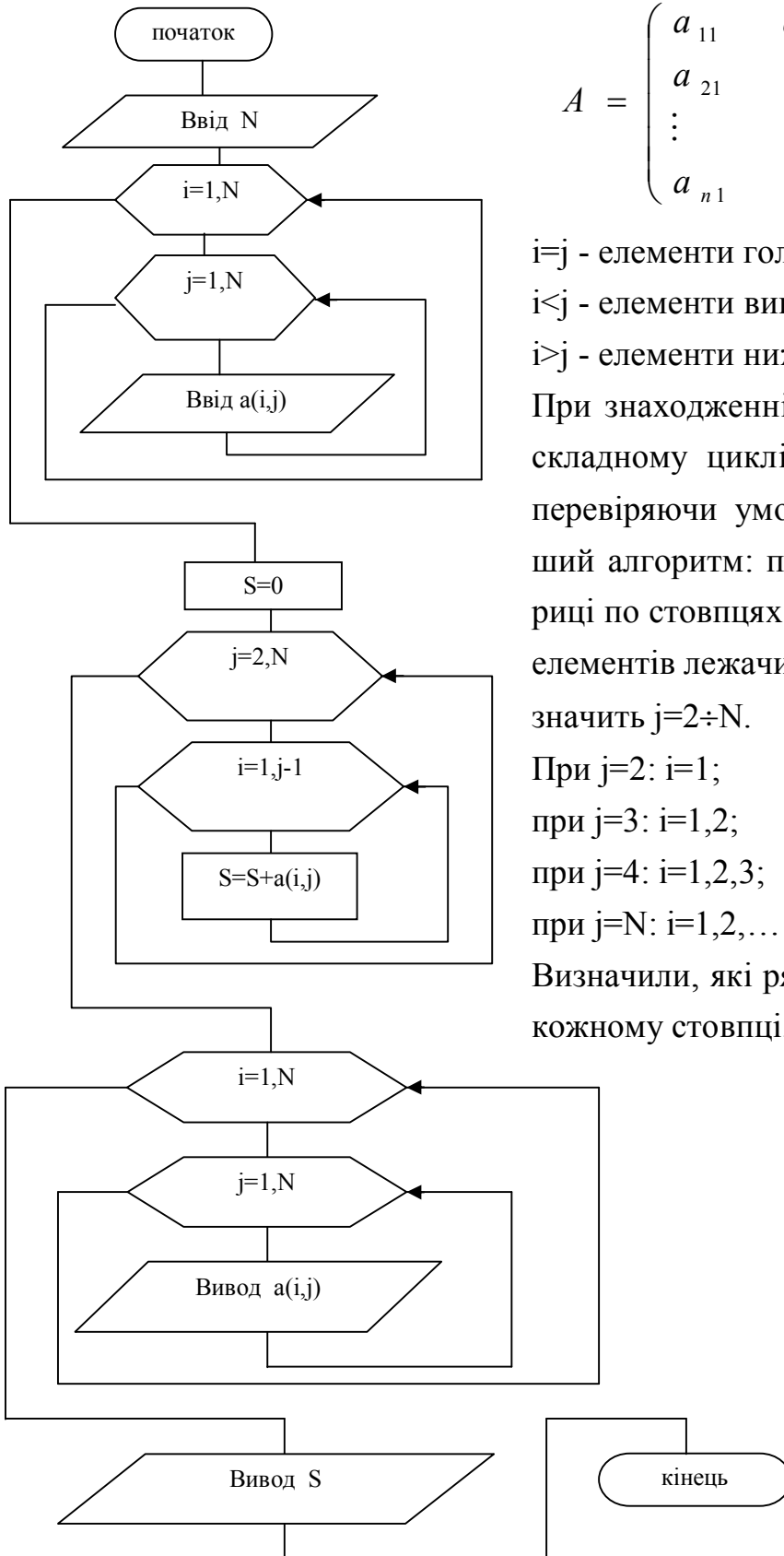
$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

$i=j$ - елементи головної діагоналі;
 $i<j$ - елементи вище головної діагоналі;
 $i>j$ - елементи нижче головної діагоналі.

При знаходженні суми елементів, можна в складному циклі перебрати всі елементи, перевіряючи умову $i<j$. Ми розберемо інший алгоритм: перебираємо елементи матриці по стовпцях. У першому стовпці немає елементів лежачих вище головної діагоналі, значить $j=2 \div N$.

При $j=2$: $i=1$;
 при $j=3$: $i=1,2$;
 при $j=4$: $i=1,2,3$;
 при $j=N$: $i=1,2,\dots, N-1$.

Визначили, які рядки будуть оброблятися в кожному стовпці.



Приклад №4. Поміняти місцями елементи двох рядків матриці $A(M,N)$.

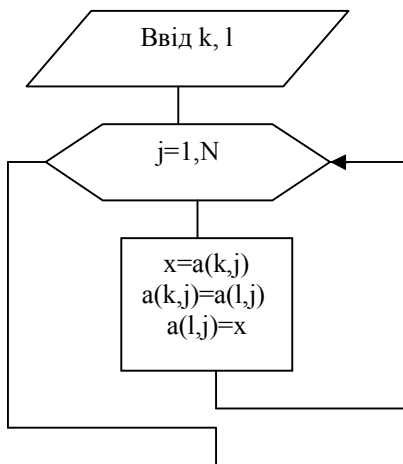
Номери рядків задати.

Розглянемо алгоритм перестановки елементів рядків матриці:

Припустимо, що потрібно поміняти місцями елементи 2-ий і 4-ої рядка, тобто потрібно поміняти місцями: a_{21} і a_{41} , a_{22} і a_{42} , ..., a_{2N} і a_{4N}

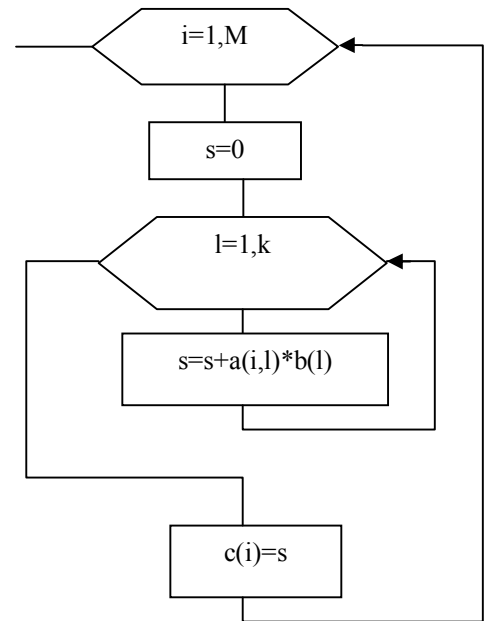
$x=a_{21}$ - запам'ятовуємо в додаткової змінної того типу, що й елементи масиву.

$a_{21}=a_{41}$, $a_{41}=x$.



1. Множення матриці на вектор - це є вектор $(N=1)$.

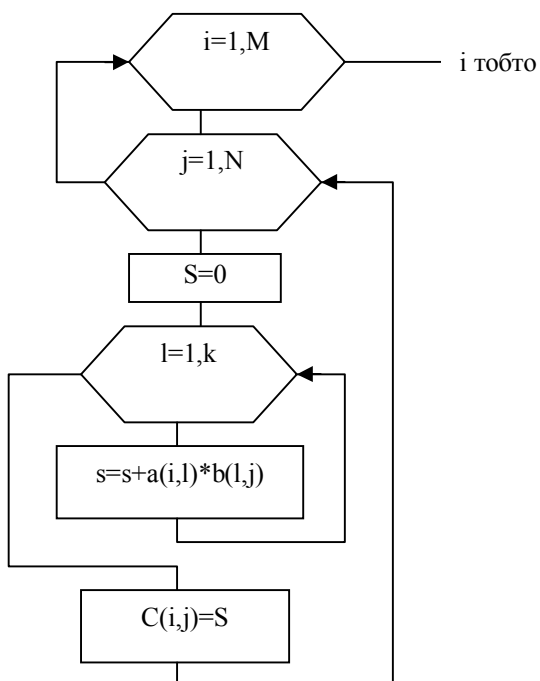
і тобто



2. Добуток матриць: $(m,k) \cdot (k,n) = (m,n)$.

$$A(M,K) \cdot B(K,N) = C(M,N).$$

Якщо $C=A \cdot B$, то $c_{ij} = \sum_{l=1}^k a_{il} \cdot b_{lj}$, где $i = \overline{1, M}$; $j = \overline{1, N}$



Розглянемо докладніше цей випадок:

Нехай дано:

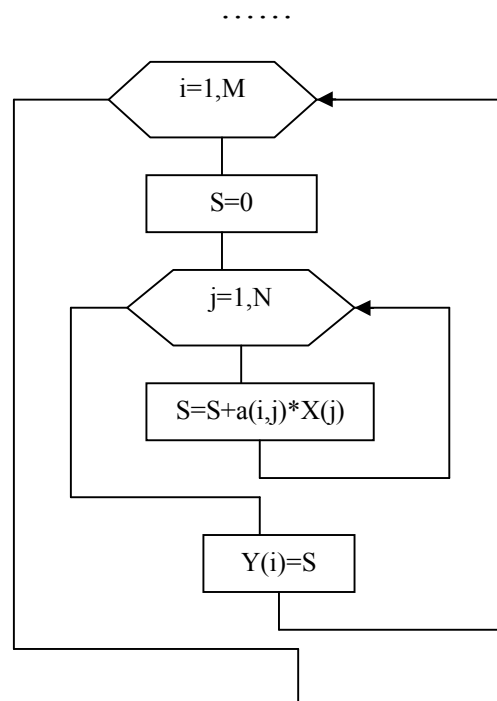
$$A(M,N) = \{a_{ij}\}_{i=1, j=1}^{M, N} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

$$X(N) = \{x_i\}_{i=1}^N = \{x_1, x_2, \dots, x_N\}$$

$$A \cdot X = Y, \text{ де } Y(M) = \{y_i\}_{i=1}^M = \{y_1, y_2, \dots, y_M\}$$

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix}$$

$$(M,N) \cdot (N,1) = (M,1), \quad y_i = \sum_{j=1}^N a_{ij} x_j, \quad \text{де } i = \overline{1, M}$$



Лабораторна робота №8.

Тема: «Обробка двовимірних масивів»

Самостійна робота. Виконати транспонування матриці (поміняти місцями рядки й стовпці матриці $A(N,N)$).

Лекція №10. ОСНОВИ РОБОТИ В MS EXCEL

Табличний редактор MS Excel призначений для обробки таблиць, виконання обчислень і побудови діаграм. Робочий файл редактора називається *Книга* й може складатися з одного або декількох *Аркушів* (рис. 1). Кожен аркуш являє собою таблицю, стовпці якої позначаються латинськими літерами, рядки – цифрами. На перетинанні стовпця й рядка знаходиться *Комірка*. Кожен комірка має свою унікальну *Адресу* (наприклад: A1, AB345 і т.д.). Адреса поточної комірки завжди показана у полі Ім'я.

Декілька рядом розташованих комірок називаються *Діапазоном*, що може бути прямокутним, у вигляді рядка або стовпця. Адреса діапазону складається з адрес лівої верхньої і правої нижньої комірок, які записані через : (наприклад, A2:C4, A3:E3, B1:B10 і тобто). При виділенні декількох несуміжних діапазонів необхідно тримати натиснутою клавішу Ctrl.

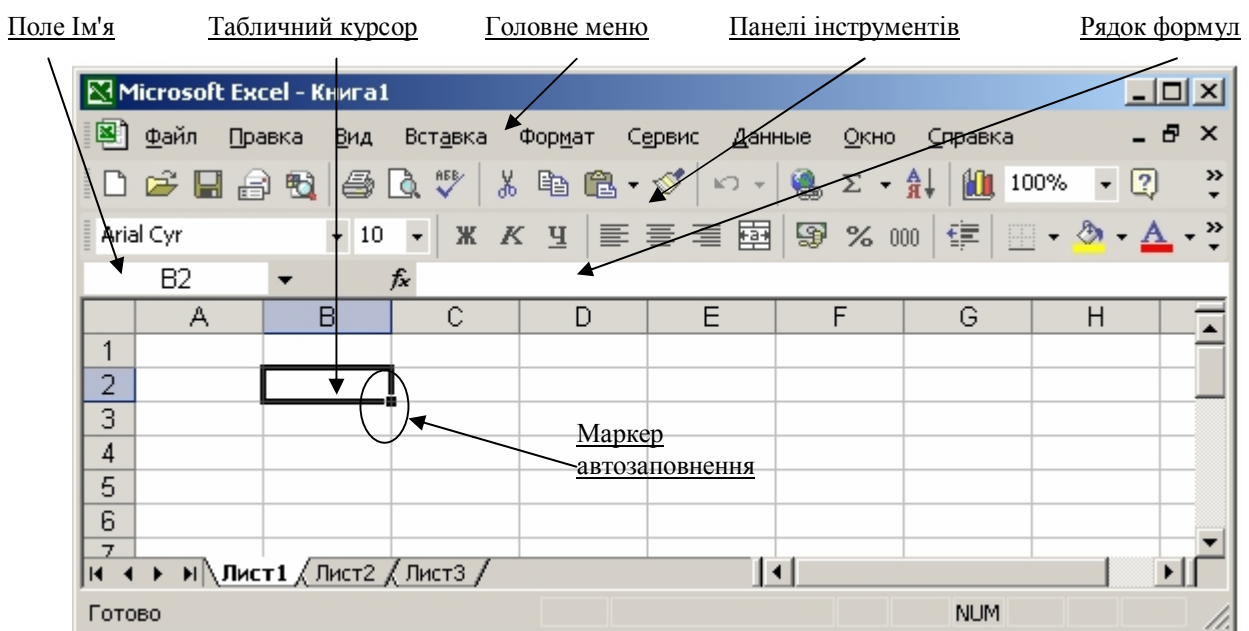


Рисунок 1. Вікно редактора MS Excel.

10.1. Ввід і коректування даних, форматування комірок

Для вводу даних необхідно поставити табличний курсор у необхідну комірку і ввести значення із клавіатури. Дані бувають наступних типів: текст, числа, дати, формули й об'єкти.

Для заповнення діапазону комірок однаковими величинами необхідно: ввести в 1-у комірку необхідне значення, поставити курсор миші на *Маркер автозаповнення* (рис. 1), він прийме вигляд \dagger , і розтягти область виділення на весь діапазон. Аналогічним способом заповнюються діапазони комірок, що містять величини, які перераховуються, тобто назви днів тижня й місяців, нумеровані списки (1-й клас, 2-й клас і тобто). Для заповнення комірок зростаючими послідовностями числових значень можна скористатися командою **Правка – Заповнить – Прогрессія**, перед викликом якої необхідно ввести 1-е значення послідовності й виділити заповнюваний діапазон.

Для коректування вмісту комірок можна або натиснути клавішу F2 і внести зміни в самої комірці, або клацнути лівою кнопкою мишки в рядку формул і виконати там редагування даних.

При введенні тексту він може перекривати розташовані праворуч комірки, які насправді залишаються вільними. Тому при створенні таблиць заголовков кожного стовпця необхідно вводити не в першу «вільну» праворуч комірку, а в наступну після поточної, щоб у таблиці не залишалось порожніх стовпців. І тільки після введення всієї «шапки» таблиці рекомендується переходити до її форматування.

Під форматуванням розуміється зміна зовнішнього вигляду комірок і їхнього вмісту. Для цього необхідно виділити діапазон комірок (якщо форматується одна комірка, то в неї досить поставити табличний курсор), і виконати команду **Формат – Ячейки...** Вікно «Формат комірок» (рис. 2) містить наступні вкладки:

Число – дозволяє встановлювати різні числові формати відображення даних;

Вирівнювання – дозволяє вирівнювати вміст комірок по вертикалі й горизонталі, змінювати орієнтацію тексту, поєднувати комірки й розташовувати текст комірки в кілька рядків (режим переносити за словами);

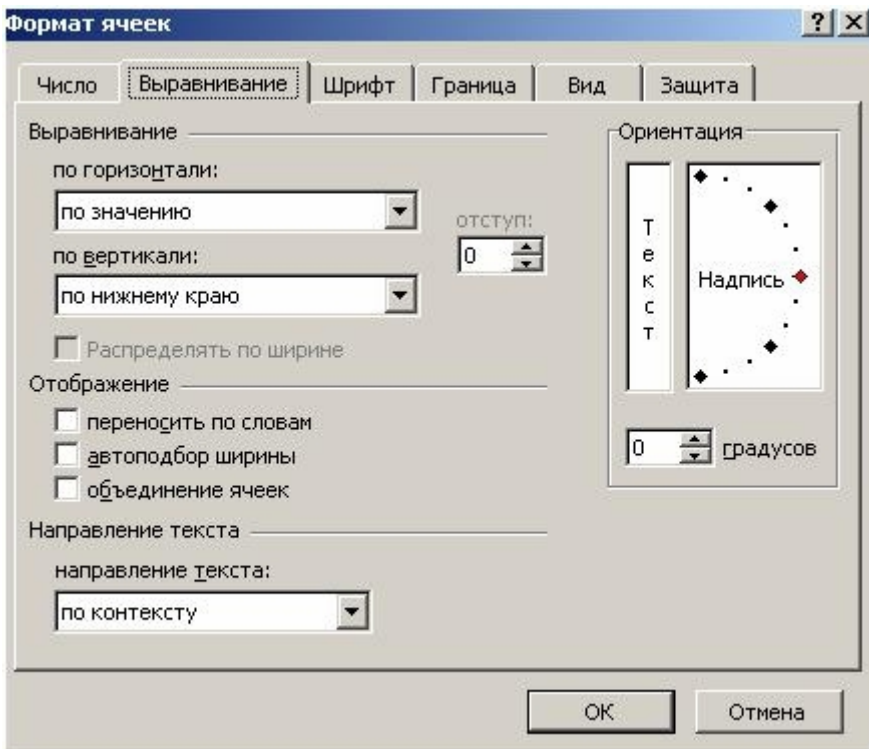


Рисунок 2. Вікно «Формат комірок»

змінювати параметри шрифту (тип, розмір, накреслення, ефекти);

Шрифт – дозволяє змінювати параметри шрифту (тип, розмір, накреслення, ефекти);

Границя – дозволяє обводити комірки рамками;

Вид – дозволяє виконувати заливання комірок різними кольорами.

10.2. Використання вбудованих функцій для обчислень

Обчислення в MS Excel виконуються за допомогою **формул**, які починаються зі знака =. Як аргументи формул можуть використовуватися числа, адреси комірок і функції. Наприклад, $=A5*C5/(B5^2)$, (вміст комірки A5 помножити на вміст комірки C5 і розділити на вміст комірки B5, яке піднесене в ступінь 2). При створенні формули адреса комірки не вводиться із клавіатури, досить лише клацнути лівою кнопкою миші на необхідній комірці й посилання на неї з'явиться у виразі.

Для обчислень за допомогою стандартних функцій MS Excel необхідно: поставити табличний курсор в комірку, у якій потрібно отримати результат; виконати команду **Вставка – Функція**, або натиснути на панелі інструментів кнопку *fx* для виклику **Майстра функцій**. Майстер – це послідовність діалогових вікон, у кожному з яких необхідно вибрати або ввести необхідні параметри, для

одержання кінцевого результату. На 1-му кроці Майстра функцій необхідно вибрати категорію функції (для зручності використання, функції розділені на категорії), знайти й виділити функцію, потім натиснути Ок. Якщо невідомо категорію, до якої відноситься функція, то спочатку рекомендується переглянути категорію «10, що недавно використовувалися», а потім «Повний алфавітний перелік». Функції, які найбільш часто використовуються, приведені в табл. 10.1.

Таблиця 10.1

Функції MS Excel, які найбільш часто використовуються.

№ п/п	Функція	Категорія	Призначення
1	СУММ(число1; ...)	Математичні	Обчислює суму значень діапазону комірок.
2	СРЗНАЧ(число1; ...)	Статистичні	Обчислює середнє значення діапазону комірок.
3	МАКС(число1; ...)	Статистичні	Знаходить максимальне значення в діапазоні комірок.
4	МИН(число1; ...)	Статистичні	Знаходить мінімальне значення в діапазоні комірок.
5	СЧЕТ(знач1; ...)	Статистичні	Підраховує кількість чисел у діапазоні комірок.
6	СУММПРОИЗВ(мас1; ...)	Математичні	Перемножує відповідні елементи заданих масивів (діапазонів) і повертає суму добутків.
7	СУММКВ(число1; ...)	Математичні	Обчислює суму квадратів значень діапазону комірок.

На 2-му кроці Майстра функцій необхідно виділити адреси комірок або діапазонів, які будуть аргументами функції й натиснути Ок. Якщо це вікно закриває дані на аркуші, те його можна відсунути убік мишею.

Для обчислення суми значень діапазону комірок зручно виконати наступні дії: поставити табличний курсор в комірку, у якої необхідно отримати результат;

натиснути кнопку **Автосума** (Σ) на панелі інструментів; виділити необхідні комірки й натиснути Enter.

При виконанні обчислень аналогічні формули не створюють заново, а отримують за допомогою автозаповнення.

В MS Excel існують три види адресації осередків - відносна, абсолютна й змішана.

У випадку **відносної адресації** адреса комірки задається звичайним способом (наприклад, A3, C12 і тобто). При копіюванні формул за допомогою маркера автозаповнення, відносні адреси які містяться в них, будуть змінюватися. Якщо формула «розтягується» по рядку, то в адресі буде змінюватися номер стовпця. Якщо по стовпцю, то номер рядка.

У випадку **абсолютної адресації** у адресі комірки перед літерою й цифрою додається знак \$ (наприклад, \$A\$3, \$C\$12 і тобто). При копіюванні формул за допомогою маркера автозаповнення, абсолютні адреси, які містяться в них, змінюватися не будуть. Виходить, якщо в аналогічних формулах повинна стояти той сама адреса, то її треба зробити абсолютною. Для цього після вводу адреси у формулу необхідно натиснути F4 (на адресу додадуться знаки \$).

У випадку **змішаної адресації** знак \$ додається перед літерою або перед цифрою в адресі комірки (наприклад, \$A3, C\$12 і тобто). При копіюванні формули за допомогою маркера автозаповнення не буде змінюватися номер стовпця або рядка в змішаній адресі, перед яким стоїть знак \$.

Приклад. Обчислити функцію $Y = \text{Cos}(\pi/2 \cdot X)/A$, при $X \in [0;1]$, $hX=0,2$. Для цього в діапазон A2:A12 за допомогою **Правка – Заповнить – Прогрессія** занесемо значення X із заданого інтервалу. В комірку B2 введемо формулу для обчислення Y (формула приведена в комірни C2). Для обчислення наступних значень Y у стовпці B будуть використовуватися формули аналогічні, уведеної в комірку B2. При цьому адреса комірки, що містить значення X, повинна змінюватися в кожній наступній формулі (використовується відносна адреса), а адреса комірки, яка містить значення A, повинна залишатися незмінною в кожній наступній формулі (використовується абсолютна адреса).

	A	B	C	D
1	X	Y (значення)	Y (формули)	A
2	0	0,8333	=COS(ПИ()/2*A2)/\$D\$2	1,2
3	0,2	0,7925	=COS(ПИ()/2*A3)/\$D\$2	
4	0,4	0,6742	=COS(ПИ()/2*A4)/\$D\$2	
5	0,6	0,4898	=COS(ПИ()/2*A5)/\$D\$2	
6	0,8	0,2575	=COS(ПИ()/2*A6)/\$D\$2	
7	1	0	=COS(ПИ()/2*A7)/\$D\$2	

У ході обчислень часто доводиться вибирати одне з декількох значень залежно від умов. Для цих цілей можна використовувати функції з категорії «Логічні».

Функція И (логічне І) має наступний синтаксис:

=И(логічне_значення1; логічне_значення2; ...)

Логічне_значення1, логічне_значення2, ... — це від 1 до 30 умов, що перевіряють, які можуть мати значення або ІСТИНА, або НЕПРАВДА.

Функція ИЛИ (логічне АБО) має наступний синтаксис:

=ИЛИ(логічне_значення1; логічне_значення2; ...)

Функція НЕ (логічне заперечення). Синтаксис написання:

=НЕ(логічне_значення1; логічне_значення2;...)...

Змінює на протилежне логічне значення свого аргументу. Функція **НЕ** використовується в тих випадках, коли необхідно бути впевненим у тому, що значення не дорівнює деякій конкретній величині.

Функція ЕСЛИ має наступний синтаксис:

=ЕСЛИ(лог_вираз; значення_якщо_істина; значення_якщо_неправда)

Лог_вираз — це будь-яке значення або вираз, що приймає значення ІСТИНА або НЕПРАВДА.

Значення_якщо_істина — це значення, що повертається, якщо лог_вираз дорівнює ІСТИНА. Якщо лог_вираз дорівнює ІСТИНА, а значен-

ня_якщо_істина порожньо, то повертається значення 0. Значення_якщо_істина може бути формулою.

Значення_якщо_неправда — це значення, що повертається, якщо лог_вираз дорівнює НЕПРАВДА. Якщо лог_вираз дорівнює НЕПРАВДА, а значення_якщо_неправда опущена (тобто після значення_якщо_істина немає крапки з комою), те повертається логічне значення НЕПРАВДА. Якщо лог_вираз дорівнює НЕПРАВДА, а значення_якщо_неправда порожньо (тобто після значення_якщо_істина стоїть крапка з комою з наступною закриваючою скобкою), те повертається значення 0. Значення_якщо_неправда може бути формулою.

Функції ЕСЛИ можуть бути вкладені друг у друга як значення аргументів значення_якщо_істина й значення_якщо_неправда для конструювання більш складних перевірок.

10.3 Лабораторна робота №9

Тема: Використання убудованих функцій Excel.

Організація вкладених циклів в Excel.

1. Обчислити значення функції, за допомогою убудованих функцій Excel ЕСЛИ й И.

$$y = \begin{cases} 2x, & \text{якщо } -1 \leq x < 0 \\ x^2, & \text{якщо } 0 \leq x \leq 1 \\ -x + 1, & \text{якщо } 1 < x \leq 2 \end{cases} \quad \text{при } hx = 0,1$$

	А	В
1	x	y
2	-1	-2
3	-0,5	-1
4	0	0
5	0,5	0,25
6	1	1
7	1,5	-0,5
8	2	-1

=ЕСЛИ(И(A2≥-1; A2<0); 2*A2; ЕСЛИ(И(A2≥0; A2≤1); A2^2; -A2+1))

Виконати команду:

Правка – Заполнить – Прогрессия

2. Обчислити значення функції: $Z=x^2 y - x$, при $-2 \leq x \leq 4$, $hx=1$

$$-2 \leq y \leq 2, h_y=1$$

	A	B	C	D	E	F	G	H
1	y \ x	-2	-1	0	1	2	3	4
2	-2	-6	-1	0	-3	-10	-21	-36
3	-1	-2	0	0	-2	-6	-12	-20
4	0	2	1	0	-1	-2	-3	-4
5	1	6	2	0	0	2	6	12
6	2	10	3	0	1	6	15	28

=B1^2*A2 - B1 – для комірки

=B1^2*\$A2 - B1 – по рядку

=B\$1^2*\$A2 – B\$1 – по стовпцю.

Формула вводиться в комірку B2.

10.4. Макрос

Макрос – це один раз виконана й збережена послідовність будь яких дій, (форматування зовнішнього вигляду комірок, ввід даних, обчислення, копіювання й тобто).

Макрос являє собою програму на Visual Basic. Створений макрос можна багаторазово виконувати. Щоб макрос працював не з однією поточною коміркою, а з будь-яким діапазоном комірок необхідно перед створенням макросу виділити діапазон комірок, виконати команду: **Сервіс – Макрос – Начать запись**, потім у вікні, що з'явилося, ввести ім'я макросу, задати комбінацію клавіш для швидкого виклику макросу й натиснути кнопку Ok. Всі подальші дії, що виконуються, будуть записуватися в макрос. Після виконання необхідних операцій виконати команду: **Сервіс – Макрос – Остановить запись**.

Для застосування макросу виділити діапазон комірок для якого треба виконати макрос, виконати команду: **Сервіс – Макрос – Макросы**.

У вікні, що з'явилося (рис. 3), виділити ім'я необхідного макросу, натиснути кнопку **Виконати**. При відкритті файлу, що містить макроси, буде з'являтися вікно - попередження про те, що у файлі є макроси, натиснути кнопку не відключати макроси.

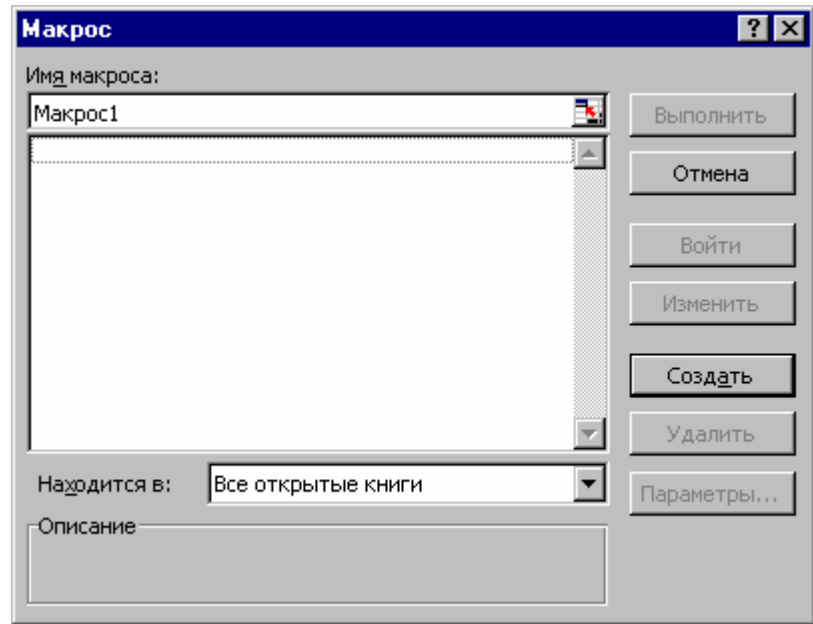


Рисунок 3. Запуск макросу на виконання

Лекція №11. VISUAL BASIC FOR APPLICATION (VBA)

11.1. Загальні зведення про об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) являє собою методику аналізу, проектування й написання додатків за допомогою об'єктів - фрагментів коду, що володіють властивостями й методами. Об'єкти створюються на базі класів і можуть моделювати правила обробки даних, різні ситуації й фізичні предмети.

Самим фундаментальним об'єктом **VISUAL BASIC** є об'єкт **форма**, завдяки якій додатки є візуальними. Форми мають множину властивостей, які є відкритими й можуть вільно змінюватися програмами. Форми мають також множину методів і подій.

Таким чином, ООП дає можливість користувачу, що створює додаток, розподілити функції програми по декількох незалежних об'єктах. Оптимізація об'єктів знижує ризик небажаної взаємодії фрагментів програми.

Властивості описують об'єкт, кожний створюваний об'єкт має ім'я класу, по якому його можна відрізнити від інших об'єктів.

Методами називаються дії, які виконуються об'єктом. Наприклад, об'єкт - вікно може відображати або приховувати себе на екрані.

Спадкуванням називається здатність об'єкта зберігати атрибути класу – батька. Щоб створити екземпляр об'єкта в VISUAL BASIC, варто вказати на його приналежність деякому класу й потім скористатися ключовим словом New. Новий об'єкт успадковує властивості й методи батька.

Інкапсуляцією називається механізм, завдяки якому дані й методи об'єкта приховуються від зовнішнього миру. Інкапсуляція запобігає сторонньому доступу до конфіденційної інформації. В VISUAL BASIC інкапсуляцію забезпечує ключове слово Private.

Поліморфізм - здатність об'єкта приймати різні форми. Поліморфізм дозволяє додавати, видозмінювати або видаляти деякі особливості поведінки до вільного об'єкта.

11.2. Інтегрована середовище розробки додатків мовою VBA

Для виклику інтегрованої середовища розробки додатків (IDE) необхідно виконати команду: **Сервіс – Макрос – Редактор Visual Basic**. IDE складається з декількох компонентів: головного меню, панелі інструментів, вікна проекту, вікна властивостей, панелі елементів, конструктора форм, вікна контрольних значень і декількох інших допоміжних вікон.

Головне меню – це рядок тексту, розташований у верхній частині вікна Visual Basic, і складається з декількох пунктів.

Меню Файл призначений для роботи з файлами, з яких утворюються додатки. У ньому можна створювати, зберігати й друкувати проекти.

Меню Правка виконує стандартні операції з буфером обміну – вирізання, копіювання й вставка. Вони застосовуються не тільки до фрагментів програми, але й до керуючих елементів.

У **Меню Вид** включаються режими перегляду різних компонентів і інструментів. Переглядати можна форми й програмні модулі.

Меню Вставка дозволяє додавати процедури, форми, модулі й модулі класу.

Команди **Меню Формат** визначають розташування й розміри елементів і форм.

За допомогою команд **Меню Отладка** можна запустити й зупинити додаток, розставити точки переривання й вибрати для перегляду об'єкти, а також виконати інші операції, що допомагають стежити за роботою додатка.

Команди **Меню Запуск** запускають і зупиняють додаток, переривають і відновляють виконання програми, що особливо зручно в процесі налагодження.

Меню Сервис дозволяє включити додаткові елементи, запустити макроси й настроїти параметри редактора.

Меню Окно дозволяє розташувати вікна IDE (каскадне або мозаїчне розташування), упорядкувати значки згорнутих форм, а також створює список, що дозволяє швидко перейти до одного з відкритих вікон IDE.

Меню ? - допомога користувачу.

Для швидкого виклику головного меню необхідно натиснути F10.

Панель інструментів знаходиться під головним меню. Якщо вона відсутня, необхідно виконати команду: **Вид – Панели инструментов – Standart.**

Вікно проекту нагадує собою вікно провідника Windows, і призначено для швидкого перегляду складових проекту, що об'єднує в собі всі об'єкти, які складають додаток.

Вікно властивостей відображає різні атрибути виділеного об'єкта. Всі об'єкти (форми, елементи, що управляють й тобто) мають атрибути, які змінюють не тільки зовнішній вигляд об'єкта, але і його поведінку. Всі ці атрибути називаються властивостями. Отже, кожний об'єкт має набір властивостей.

Вікно контрольних значень дозволяє переглядати значення контрольних змінних у процесі перевірки правильності роботи (налагодження) проекту, що дозволяє знаходити помилки в логіці роботи програм.

Конструктор форм розташований у центрі екрана редактора VBA. Тут виводиться або зображення форми, що дозволяє робити візуальне конструювання

макета форми й елементів, які розташовані на ній, або вікно програми.

11.3. Особливості програмування мовою VBA.

Процес розробки програм мовою VBA складається з декількох етапів, залежно від кінцевого результату. Якщо необхідно одержати програму, яка буде робити визначені обчислення або дії, то досить створити програмний модуль, що містить текст вхідного коду. Для запуску цієї програми можна створити кнопку, при натисканні якої програма буде виконуватися. Для цього необхідно включити панель інструментів за допомогою команди: **Вид – Панелі інструментов – Елементи управління**, а потім створити кнопку. Або виконати програму за допомогою команди: **Сервис – Макрос – Макросы**.

Розробка “повноцінної” програми буде включати два етапи.

Перший етап – етап візуального програмування, на якому створюється вікно (форма) програми, де розташовуються необхідні елементи управління.

Другий етап – етап програмування у вхідному коді, на якому створюються частини програми, що виконуються у відповідь на певні події. Подією є, наприклад, щиклик лівою кнопкою миші на командній кнопці (подія Click), натискання клавіші на клавіатурі (подія KeyPress) і тобто.

11.4. Типи даних.

Всі значення, з якими працює програма, зберігаються в змінних, константах, масивах. **Змінною** називається область пам'яті, у якій можуть зберігатися різні значення. Кожна змінна в VBA має свій тип, який вказує, що може зберігати змінна: ціле число, рядок, дату й тобто (табл. 11.1).

У процесі виконання програми значення змінних можуть змінюватися за допомогою оператора присвоєння: “=”. Таким чином, розрізняють:

опис змінних (визначення їхнього типу);

ініціалізацію змінних (присвоєння ним початкових значень).

Змінну в VBA можна описати за допомогою наступної конструкції:

Dim ІМ'Я_ЗМІННОЇ **As** ТИП_ЗМІННОЇ

Наприклад:

Dim A As Integer- оголошується змінна A цілого типу;

Dim E As Single - оголошується змінна E дійсного типу;

Dim S As String - оголошується строкова змінна S.

VBA дозволяє не описувати змінні перед використанням у програмі. У цьому випадку за замовчуванням використовується тип Variant. Змінні цього типу можуть зберігати все, що в них помістять, тобто їхній тип змінюється залежно від останнього присвоєння. Однак рекомендується всі змінні, які будуть використовуватися в програмі, описувати явно. Для заборони використання змінних, які не були явно описані на початку програми, необхідно вставити рядок Option Explicit.

Таблиця 11.1.

Стандартні типи даних VBA

Типи даних	Розмір пам'яті (байт)	Призначення
1	2	3
Boolean (Логічний)	2	Містить значення True (Істина) або False (Неправда).
Byte (Позитивне ціле)	1	Містить цілі позитивні числа в діапазоні від 0 до 255.
Integer (Ціле)	2	Містить цілі числа в діапазоні від -32768 до 32768
Long (Довге ціле)	4	Містить цілі числа в діапазоні від -2147483648 до 2147483647
Single (Число із плаваючою точкою)	4	Містить речовинні числа по абсолютній величині від 1,401298E-45 до 3,402823E+38

1	2	3
Date (Дата)	8	Містить дати від 1.01.0100г. до 1.12.9999р.
Double (Число із плаваючою точкою подвійної точності)	8	Містить речовинні числа по абсолютній величині від 4,940656438E-324 до 1,79769313E+308
Object (Об'єкт)	4	Містить посилання на будь-який заданий об'єкт.
String (Строковий)	10 + довжи- на	Містить текстові або строкові значення довжиною від 0 до $2 \cdot 10^9$.
Currency (Грошовий)	8	Містить грошові величини в діапазоні від -922337203685477,58 до 922337203685477,58
Variant (Варіант)	-	Універсальний тип загального призначення, що може зберігати значення більшості інших типів.

Часто при написанні програм необхідно використовувати ті самі постійні значення: числа, рядка, дати й тобто. У цьому випадку можна задати константу за допомогою однієї з наступних конструкцій:

Const ІМ'Я_ПОСТІЙНОЇ = ВИРАЗ

або

Const ІМ'Я_ПОСТІЙНОЇ **As** ТИП_ПОСТІЙНОЇ = ВИРАЗ

Наприклад: **Const** FileName = "text.xls"

Const PI **As** Double = 3.14159

Лекція №12. УМОВНИЙ ОПЕРАТОР.

ОПЕРАТОРИ ЦИКЛУ В VBA

12.1. Оператор If

Оператор If ... Then ... Else являє собою найпростішу форму перевірки умов в VBA, і має наступний синтаксис:

If УМОВА **Then** Оператор_1 **Else** Оператор_2

Оператор_1 виконується, якщо УМОВА істинно, у протилежному випадку виконується Оператор_2.

При цьому оператор If ... Then ... Else записують в один рядок.

Умова - це вираз логічного типу. Результат виразу завжди має булевий тип. В умовному операторі припустиме використання блоку операторів замість кожного з операторів.

У цьому випадку умовний оператор має вигляд:

```
If УМОВА Then
    Блок_операторів_1
Else
    Блок_операторів_2
End If
```

Гілка Else в умовному операторі є необов'язковою.

В операторі умови після Then можна розмістити декілька операторів, для того, щоб всі вони виконувалися, якщо умова істинно. У цьому випадку вони повинні розташовуватися в один рядок і бути розділені двокрапкою, наприклад:

```
If A>10 Then A=A+1: V=B+A : C=C+V
```

В умовному операторі може перевірятися кілька умов. У цьому випадку умовний оператор має вигляд:

```
If УМОВА_1 Then
    Блок_операторів_1
Else If УМОВА_2 Then
    Блок_операторів_2
```



```

Else
...
End If
End If

```

12.2. Оператори циклу

Для організації циклу “Для” в VBA служить оператор For...Next, що має наступний синтаксис:

```

For Лічильник=поч_значення To кін_значення Step Крок
    Блок_операторів
[Exit For]
Next лічильник

```

В VBA для організації циклів з невідомим числом повторень використовуються оператори циклу: **Do While...Loop**, **Do Until...Loop**, **Do...Loop While**, **Do...Loop Until...**

Do While УМОВА	Do
Блок_операторів	Блок_операторів
[Exit Do]	[Exit Do]
Блок_операторів	Блок_операторів
Loop	Loop While УМОВА

Do Until УМОВА	Do
Блок_операторів	Блок_операторів
[Exit Do]	[Exit Do]
Блок_операторів	Блок_операторів
Loop	Loop Until УМОВА

Оператор **Do While...Loop** забезпечує багаторазове повторення блоку операторів доти, поки УМОВА дотримується, а оператор **Do Until ... Loop** поки УМОВА не дотримується. Оператори **Do...Loop While**, **Do...Loop Until** відрі-

няються від перерахованих вище операторів тим, що спочатку блок операторів виконується, принаймні, один раз, а потім перевіряється УМОВА. Для запобігання зациклення в тілі циклу повинен бути хоча б один оператор, який змінює значення змінних, які присутні в УМОВІ. Оператор **Exit Do** забезпечує достроковий вихід з оператора циклу.

12.3. Лабораторна робота №2.

Тема: «Організація циклів з відомим числом повторень»

Приклад №1. Скласти програму для обчислення $y = \frac{\sin ax + 2}{(a + x)^3}$,

для $3 \leq x \leq 6$; $hx=0,8$. Обчислити k - в $Y < 0,3$; $S = \Sigma y$, $P = \Pi y$.

Ввід x_n , x_k , h_x , a за допомогою текстових полів форми.

P , S , k – вивести на форму. X , Y - вивести на аркуш Excel.

Dim a As Single, x As Single

Dim y As Single, hx As Single

Dim p As Single, s As Single

Dim xn As Single, xk As Single

Dim k As Integer, i As Integer

*'Очищення всіх текстових полів
при натисканні кнопки "Очищення"*

Sub CommandButton2_Click()

TextBox1=" ": TextBox2=" ":

TextBox3=" ": TextBox4=" ":

TextBox5=" ": TextBox6=" ":

TextBox7=" "

End Sub

'Завершення роботи програми при

'натисканні кнопки " Вихід"

Sub CommandButton3_Click()

End

End Sub

'Розрахунок необхідних величин при натисканні кнопки " Обчислення"

Sub CommandButton1_Click()

xn=CSng(TextBox1): xk=CSng(TextBox2): hx=CSng(TextBox3): a=CSng(TextBox4)

k=0: p=1: s=0: i=2: x=xn

Cells(1,1)= " x" : Cells(1,2)= " y"

Do While x≤xk

Y=(sin(a*x)+2)/(a+x)^3

Cells(i,1)=x: Cells(i,2)=y

If y<0.3 then k=k+1

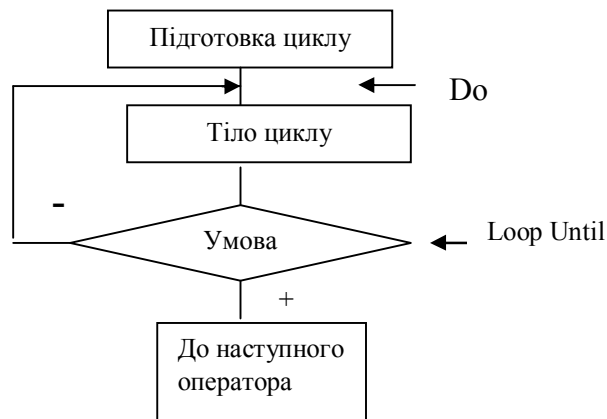
s=s+y : p=p*y : i=i+1

x=x+hx

Loop

TextBox5=CStr(p): TextBox6=CStr(s): TextBox7=CStr(k)

End Sub



Лекція №13. АРИФМЕТИЧНІ ОПЕРАТОРИ. СТАНДАРТНІ ФУНКЦІЇ В VBA. СТВОРЕННЯ МОДУЛІВ. ПРОЦЕДУРИ І ФУНКЦІЇ КОРИСТУВАЧА

13.1. Запис виразів

Оператор **присвоювання** має наступний синтаксис:

ІМ'Я_ЗМІННОЇ = ВИРАЗ

ВИРАЗ – це будь-які арифметичні (табл. 13.1) й логічні оператори, оператори порівняння й конкатенації, які написані за правилами VBA. Результат обчислення виразу заноситься в зазначену змінну. Виконувати присвоювання можна тільки між сумісними змінними.

Таблиця 13.1

Арифметичні оператори

Оператор	Опис
Оператор ^	Зводить число в ступінь.
Оператор *	Повертає добуток двох чисел.
Оператор /	Повертає результат ділення двох чисел.
Оператор \	Повертає результат цілого ділення двох чисел.
Оператор Mod	Повертає остача при цілому діленні двох чисел.
Оператор +	Повертає суму двох чисел.
Оператор –	Повертає різниця двох чисел, або змінює знак.

Якщо вираз містить декілька операторів, то значення компонентів виразу розраховуються в визначеному порядку, що називають пріоритетом операторів.

Якщо вираз містить оператори різних типів, то першими виконуються арифметичні операції, слідом за ними операції порівняння, а останніми логічні операції. Всі оператори порівняння мають рівний пріоритет, тобто виконуються в порядку їхнього розташування у виразі ліворуч праворуч. Арифметичні й логічні оператори виконуються в порядку їхнього пріоритету (табл. 13.2).

Пріоритет операторів

Арифметичні	Порівняння	Логічні
Піднесення в ступінь (^)	Дорівнює (=)	Not
Зміна знака (-)	Не дорівнює (<>)	And
Множення й ділення (*,/).	Менше (<)	Or
Ціле ділення (\).	Більше (>)	Xor
Ділення по модулю (Mod).	Менше або дорівнює (<=)	
Додавання й вирахування (+,-).	Більше або дорівнює (>=)	

Розташовані поруч у виразі оператори множення й ділення виконуються ліворуч праворуч. У такому ж порядку виконуються розташовані поруч оператори додавання й вирахування. Оператори усередині круглих скобок завжди виконуються раніше, ніж оператори поза скобками. Порядок виконання операторів, що стоять усередині скобок, обумовлюється старшинством операторів.

Наприклад, математичний вираз:
$$y = \frac{x^2 - 3 \cdot x^{0.7}}{(0.2 \cdot x + 1) \cdot (x - 4)}$$

на VBA буде мати такий вигляд: $y=(x^2-3*x^{(0.7)})/((0.2*x+1)*(x-4))$

У VBA використовуються наступні стандартні функції (табл. 13.3).

Таблиця 13.3

Стандартні функції VBA

Функція	Опис
1	2
CDate(вираз)	Перетворює вираз в тип Date
CInt(вираз)	Перетворює вираз в тип Integer
CLng(вираз)	Перетворює вираз в тип Long
CSng(вираз)	Перетворює вираз в тип Single
CStr(вираз)	Перетворює вираз в тип String
abs(аргумент)	Повертає значення, тип якого збігає з типом переданого аргументу, яке рівне абсолютному значенню указанного числа.

1	2
atn(аргумент)	Повертає значення типу Double, що містить арктангенс числа.
cos(аргумент)	Повертає значення типу Double, що містить косинус кута.
int(аргумент)	Повертає значення типу, що збігає з типом аргументу, яке містить цілу частину числа.
log(аргумент)	Повертає значення типу Double, що містить натуральний логарифм числа.
exp(аргумент)	Повертає значення типу Double, що містить результат піднесення числа e в указаний ступінь.
sin(аргумент)	Повертає значення типу Double, що містить синус кута.
sqrt(аргумент)	Повертає значення типу Double, що містить квадратний корінь числа.
tan(аргумент)	Повертає значення типу Double, що містить тангенс кута.

13.2. Створення модулів. Процедури й функції користувача

Модуль являє собою текстовий ASCII-файл із програмним кодом, що містить підпрограми, змінні й константи.

Проект може складатися із безлічі програмних модулів. Для їхнього створення необхідно виконати команду **Вставка – Модуль**.

Основу програмного коду в VBA складають процедури й функції.

Процедура Sub – це відособлена сукупність операторів VBA, що виконує визначені дії. Процедура приймає деякі параметри (змінні, які передаються процедурі в якості вхідних даних), виконує програмний код і може повертати результуючі значення, які присвоюються параметрам усередині процедури. Вкладеність процедур в інші процедури не допускається.

Структура процедури наступна:

[ДОСТУП] **Sub** ІМ'Я_ПРОЦЕДУРИ([СПИСОК_ПАРАМЕТРІВ])

*ТИЛО_ПРОЦЕДУРИ***End Sub**

Ключове слово *ДОСТУП* є необов'язковим і визначає область видимості процедури. **Public** вказує, що процедура доступна для всіх інших процедур у всіх модулях (глобальна). **Private** вказує, що процедура доступна для інших процедур тільки того модуля, у якому вона описана (локальна). *СПИСОК_ПАРАМЕТРІВ* також є необов'язковим елементом і дозволяє передавати процедурі різні вихідні дані при виклику, які називаються формальними параметрами. При цьому ключове слово **Dim** не вказується.

ТИЛО_ПРОЦЕДУРИ складається з описової частини й блоку операторів, що виконуються один за іншим. Якщо необхідно припинити виконання процедури в деякому конкретному місці, це можна зробити за допомогою оператора **Exit Sub**.

ІМ'Я ПРОЦЕДУРИ - це будь-який ідентифікатор, визначений користувачем. **Ідентифікатор** - це послідовність літер, цифр і символу підкреслення, що починається з літери (пробіли усередині ідентифікатора неприпустимі). Ім'я процедури завжди визначається на рівні модуля. Для використання процедури в тексті програми (тобто для її виклику), необхідно вказати ім'я процедури й список фактичних параметрів, які повинні по типу й порядку проходження збігатися з формальними параметрами.

Функція Function багато в чому схожа на процедуру, але на відміну від неї при виклику завжди повертає значення. Функція отримує параметри, які називані **аргументами**, і виконує з ними деякі дії, результат яких повертається функцією. Структура функції наступна:

```
[ДОСТУП] Function ІМ'Я_ФУНКЦІЇ(СПИСОК_АРГУМЕНТІВ) As ТИП  

   ТИЛО_ФУНКЦІЇ  

   ІМ'Я_ФУНКЦІЇ=ВИРАЗ
```

End Function

ТИП визначає тип даних результату, який повертається функцією. У тілі функції обов'язково повинен бути присутнім, принаймні, один оператор, що при-

своє імені функції значення виразу, який обчислюється. Дострокове завершення функції можливо за допомогою оператора **Exit Function**. У програмі виклик функції здійснюється за допомогою оператора присвоювання, у правій частині якого вказується ім'я функції з переліком фактичних параметрів, як і будь-якої іншої убудованої функції, наприклад, Sqr, Cos або Chr.

Процедури й функції, які не описані явно за допомогою ключових слів **Public** або **Private**, за замовчуванням є загальними.

Для швидкого додавання в модуль підпрограм зручно скористатися командою **Вставка – Процедура**. У вікні, що з'явилося, потрібно вибрати необхідні опції.

В MS Excel з функціями, які створені користувачем, можна працювати за допомогою *Майстра функцій* точно так само, як і з вбудованими функціями.

13.3. Лабораторна робота №3.

Тема: «Організація циклів з невідомим числом повторень»

Приклад. Скласти програму для обчислення $y = \frac{1}{3}x + \sqrt{\frac{x}{x^3 + 1}}$, $x_n=3$, $h_x=0.8$.

Обчислювати Y, поки підкореневий вираз більше 0.03. Знайти k - кількість обчислених Y. x_n , h_x ввести з форми, k - вивести на форму. X, Y - вивести на аркуш Excel.

```
Dim x As Single, y As Single, xn As Single, hx As Single
```

```
Dim k As Integer, i As Integer
```

'Очищення всіх текстових полів при натисканні кнопки "Очищення"

```
Private Sub CommandButton2_Click()
```

```
TextBox1=" ": TextBox2=" ": TextBox3=" "
```

```
End Sub
```

'Розрахунок необхідних величин при

' натисканні кнопки "Обчислення"

```
Private Sub CommandButton1_Click()
```

```
xn=CSng(TextBox1):hx=CSng(TextBox2)
```



```

k=0: i=2: x=xn
Cells(1,1)="x": Cells(1,2)="y"
Do While x/(x^3+1)>0.03
  Y=1/3*x+Sqr(x/(x^3+1))
  Cells(i,1)=x: Cells(i,2)=y
  k=k+1: i=i+1: x=x+hx
Loop
TextBox3=CStr(k)
End Sub

```



'Завершення роботи програми при натисканні кнопки " Вихід"

```

Sub CommandButton3_Click()
  End
End Sub

```

Лекція №14. ОПЕРАТОРИ ВВОДУ-ВИВОДУ В VBA. ОБ'ЄКТИ, ВЛАСТИВОСТІ Й МЕТОДИ VBA

14.1. Оператори вводу-виводу

Функція **MsgBox** здійснює вивід інформації в діалогові вікні й установлює режим очікування натискання кнопки користувачем. Вона має наступний синтаксис:

MsgBox ПОВІДОМЛЕННЯ [,КНОПКИ] [,ЗАГОЛОВОК]

Аргументи:

ПОВІДОМЛЕННЯ – обов'язковий аргумент, що задає у вікні виведене інформаційне повідомлення. Може складатися з декількох текстових рядків, об'єднаних знаком &. Використання в цьому аргументі Chr(13) приводить до переходу на новий рядок при виводі інформації.

КНОПКИ – значення цього аргументу визначає категорії кнопок, що з'являються у вікні (табл. 14.1). Від значення аргументу кнопки залежить також, чи з'являється у вікні який-небудь значок. Якщо не зазначено, які кнопки необхідно

відображати у вікні повідомлень, то використовується значення за замовчуванням, що відповідає кнопці ОК.

ЗАГОЛОВОК – задає заголовок вікна.

Функція **MsgBox** повертає значення типу Integer, що вказує, яка кнопка була натиснута в діалоговому вікні.

Таблиця 14.1

Припустимі значення змінної кнопки.

Відображення	Аргумент
Кнопка ОК	VbOKOnly
Кнопки ОК і Скасування	VbOKCancel
Кнопки Та й Немає	VbYesNo
Кнопки Так, Немає й Скасування	VbYesNoCancel
Кнопки Припинити, Повторити й Ігнорувати	VbAbortRetryIgnore
Кнопки Повторити й Скасування	VbRetryCancel
Інформаційний знак	VbInformation
Знак питання	VbQuestion
Знак вигуку	VbExclamation

Функція **InputBox** здійснює ввід значень за допомогою вікна введення й має наступний синтаксис:

ІМ'Я_ЗМІННОЇ=**InputBox**(*ПОВІДОМЛЕННЯ*[,*ЗАГОЛОВОК*])

Аргументи:

ПОВІДОМЛЕННЯ – обов'язковий аргумент. Задає у вікні інформаційне повідомлення, яке звичайно пояснює значення величини, що вводиться.

ЗАГОЛОВОК – задає заголовок вікна.

Наприклад:

MsgBox "Сума=" & CStr(S), vbOKOnly, "Результати"

виводить в окреме вікно з іменем "Результати", яке містить тільки кнопку "ОК", значення змінної S, з відповідним поясненням.

n=InputBox("Введіть розмірність масиву", " Ввід даних ")

14.2. Об'єкти, властивості й методи VBA.

Одним з основних понять VBA є об'єкт. **Об'єкт** – це те, чим Ви управляєте за допомогою програми мовою VBA, наприклад, форма, кнопка, робочий аркуш і діапазон осередків MS Excel. Кожен об'єкт має деякі **властивості**. Наприклад, форма може бути видимою або невидимою в цей момент на екрані. Інший приклад властивості об'єкта - шрифт для відображення інформації в осередку (об'єкті) робочого аркуша.

Об'єкт містить також список методів, які можуть бути до нього застосовні. **Методи** – це те, що ви можете робити з об'єктом. Наприклад, показати форму на екрані або забрати її можна за допомогою методів Show і Hide.

Таким чином, **об'єкт** – це програмний елемент, який має своє відображення на екрані, містить деякі змінні, що визначають його властивості, і деякі методи для управління об'єктом. Наприклад, в MS Excel є багато вбудованих об'єктів.

Range("Адреса") – задає діапазон (може включати тільки один осередок).

Cells(i,j) – осередок, що перебуває на перетинанні і-того рядка й j-того стовпця робочого аркуша MS Excel (i,j - цілі числа).

Sheets("Ім'я") – аркуш.

WorkSheets – робочий аркуш.

Установка значень властивостей – и це один зі способів управління об'єктами.

Синтаксис установки значення властивості об'єкта наступний:

Об'єкт.Властивість=Вираз

Основною властивістю об'єктів Cells і Range, є Value (значення), яке, однак, можна не вказувати. Наприклад:

Range("A5:A10").Value=0 – у діапазон осередків A5:A10 заноситься значення 0.

Cells(2,4).Value=n – в осередок, що перебуває на перетинанні 2-й рядка й 4-го стовпця (осередок з адресою "D2"), заноситься значення змінної n.

Синтаксис читання властивостей об'єкта наступний:

Змінна=Об'єкт.Властивість

Наприклад:

`xn=Cells(1,2).Value` або `xn=Range("B1").Value` – змінної `xn` присвоюється значення з осередку B1 поточного робочого аркуша.

Синтаксис застосування методів до об'єкта:

Об'єкт.Метод

Наприклад:

`Range("A5:A10").Clear` – очищається діапазон осередків A5:A10.

`Range("A2:B10").Select` – вибирається діапазон осередків A2:B10.

В MS Excel є багато об'єктів, причому деякі з них містять інші об'єкти. Наприклад, робоча книга містить робочі аркуші, робочий аркуш містить діапазон осередків і тобто. Об'єктом найвищого рівня є **Application** (додаток). Якщо ви змінюєте його властивості або викликаєте його методи, то результат застосовується до поточної роботи MS Excel. Наприклад:

`Application.Quit` – завершення роботи з Excel.

Відзначимо, що крапка після ім'я об'єкта може використовуватися для переходу від одного об'єкта до іншого. Наприклад:

`Application.Workbooks("Звіт").Worksheets("Травень").Rows(2).Delete`

Очищає другий рядок робочого аркуша „Травень” у робочій книзі „Звіт” .

Потрібно відзначити наступне:

- Можна не писати ім'я об'єкта `Application`, тому що це мається на увазі за замовчуванням.

- При роботі з підоб'єктом уже активізованого об'єкта немає необхідності вказувати утримуючий його об'єкт.

- VBA використовує деякі властивості й методи, які повертають об'єкт до якого вони відносяться (це дозволяє швидко вказувати потрібний об'єкт). Приклади таких властивостей: `ActiveCell` (активний осередок), `ActiveSheet` (активний аркуш), `ActiveWorkbook` (активна робоча книга). Так, установити значення активного осередку можна в такий спосіб:

ActiveCells.Value="Так".

14.3. Лабораторна робота №4

Тема: «Організація вкладених циклів»

Приклад. Скласти програму для обчислення $Z=X^2+Y^2$

при $1 \leq X \leq 2$, $h_x=0,2$; $-1 \leq Y \leq 1$; $h_y=0,5$. Вхідні дані ввести за допомогою оператора InputBox. X,Y,Z - вивести на аркуш Excel. Обчислити k_x і k_y - кількість обчислених x і y і вивести їх на аркуш Excel.

Для організації циклів використати оператор Do...Loop.

'Опис глобальних змінних

```
Dim xn As Single, xk As Single
```

```
Dim hx As Single, yn As Single
```

```
Dim yk As Single, hy As Single
```

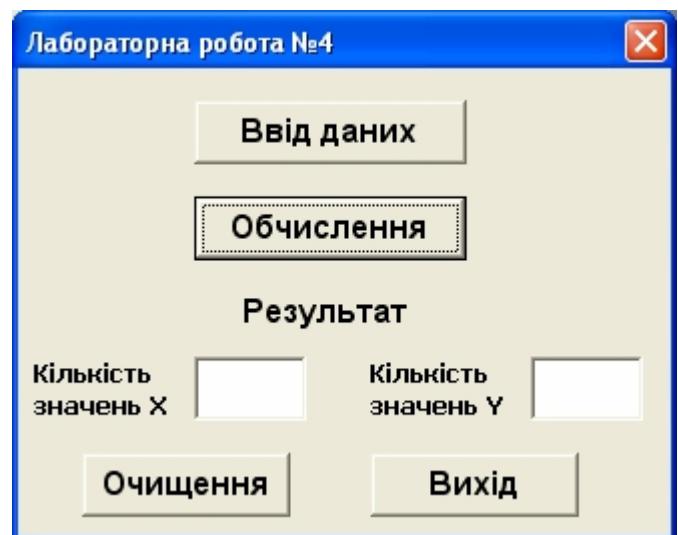
'Ввід вхідних даних при натисканні

'кнопки "Ввід даних"

```
Private Sub CommandButton1_Click()
  xn = InputBox("Ввід поч. значення x")
  xk = InputBox("Ввід кін. значення x")
  hx = InputBox("Ввід кроку по x")
  yn = InputBox("Ввід поч. значення y")
  yk = InputBox("Ввід кін. значення y")
  hy = InputBox("Ввід кроку по y")
End Sub
```

'Розрахунок z при натисканні кнопки "Обчислення"

```
Private Sub CommandButton2_Click()
  'Опис локальних змінних
  Dim x As Single, y As Single, z As Single
  Dim kx As Integer, ky As Integer, i As Integer
  Cells(1, 1) = "y": Cells(1, 2) = "x"
```



```
Cells(1, 3) = "z"
```

```
y = yn: ky = 0: i = 2
```

```
Do While y <= yk
```

```
  x = xn: kx = 0
```

```
  Do While x <= xk
```

```
    z = x ^ 2 + y ^ 2
```

```
    Cells(i, 1) = y: Cells(i, 2) = x: Cells(i, 3) = z
```

```
    kx = kx + 1
```

```
    x = x + hx
```

```
    i = i + 1
```

```
  Loop
```

```
  ky = ky + 1: y = y + hy
```

```
Loop
```

```
TextBox1 = CStr(kx): TextBox2 = CStr(ky)
```

```
End Sub
```

'Завершення роботи програми при натисканні кнопки "Обчислення"

```
Private Sub CommandButton3_Click()
```

```
  End
```

```
End Sub
```

Лекція №15.

ПРИКЛАДИ СКЛАДАННЯ ФУНКЦІЇ КОРИСТУВАЧА

Приклад. Скласти функцію користувача для обчислення функції $y=2\sin^2x$ як суми ряду:

$$2 \sin^2 x = \frac{(2x)^2}{2!} - \frac{(2x)^4}{4!} + \frac{(2x)^6}{6!} - \dots \pm \frac{(2x)^{2i}}{(2i)!} \mp \dots \quad \text{с точністю } \varepsilon=0,001$$

у точці $x=0.5$; загальний член ряду $u_i = (-1)^{i+1} \cdot \frac{(2x)^{2i}}{(2i)!}$

Для створення функції користувача виконаємо команди: **Вставка – Модуль, Вставка – Процедура**. У вікні, що з'явилося, вибрати необхідні опції.

Public Function Y(x As Single, Eps As Single) As Single

Dim S As Single, ui As Single, Fact As Single

Dim i As Integer, j As Integer

i=1: s=0

Do

Fact=1

For j=1 to 2*i

Fact=Fact*j

Next j

ui=(-1)^(i+1)*((2*x)^(2*i))/Fact

s=s+ui

i=i+1

Loop Until Abs(ui)<Eps

y=S

End Function

Для оцінки правильності роботи функції користувача обчислимо $y=2\sin^2x$ засобами Excel.

	A	B	C	D	E
1	x	0,5	Y1	=2*(sin(B1))^2	1,8391
2	Eps	0,001	Y2	=Y(B1;B2)	1,8391

Зразок оформлення курсової роботи

$$z = \begin{cases} \sqrt{x^2 + \frac{a^2}{4}} + x^3, & \text{якщо } x > 1,6 \\ a + \frac{x}{\sqrt{a-x}}, & \text{якщо } x \leq 1,6 \end{cases}, \quad x = \sqrt[3]{4+t^2} \quad \begin{matrix} 1,5 \leq a \leq 2,5; \quad ha=0,5 \\ 2 \leq t \leq 4,5; \quad ht=1 \end{matrix}$$

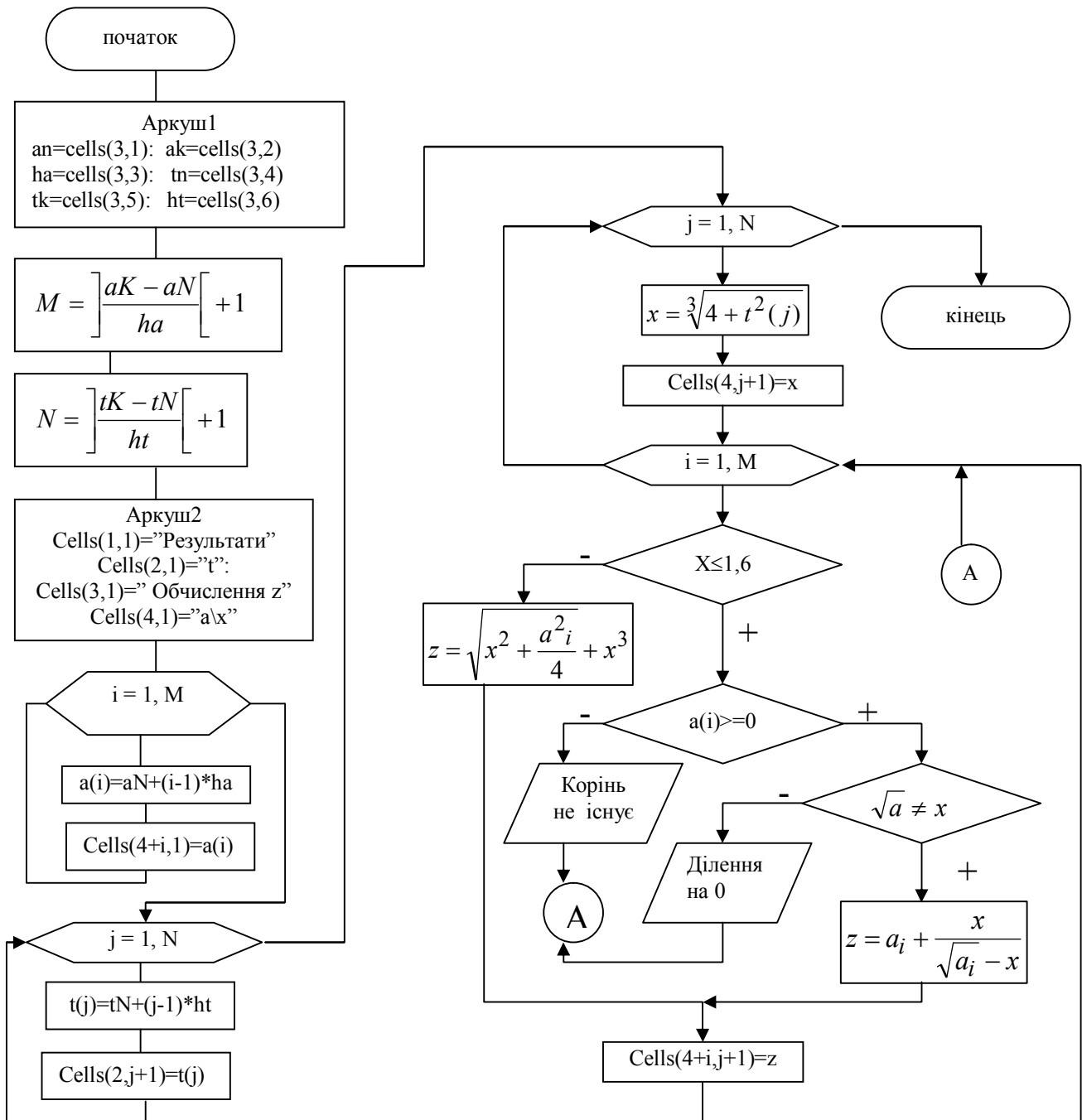
Вхідні дані вводити з 1-го аркушу Excel. Результати: a, t, x, z - вивести на 2-й аркуш Excel.

Аркуш 1

1	Вхідні дані					
2	an	ak	ha	tn	tk	ht
3	1,5	2,5	0,5	2	4,5	1

Аркуш 2

Результати					
t	
Обчислення z					
a \ x			
	z				



Лекція №16. ОПИС МАСИВІВ В VBA. ОСНОВНІ ВЛАСТИВОСТІ ФОРМ. ОСНОВНІ ПОДІЇ Й МЕТОДИ ФОРМ

16.1. Опис масивів

Масив описується наступною конструкцією:

Dim *ІМ'Я_МАСИВУ*(Розмірність) **As** *ТИП_МАСИВУ*

Наприклад:

Dim A(12) As Integer – оголошується вектор з 12 цілих чисел, причому за замовчуванням перший елемент масиву буде $A(0)$, а останній $A(11)$. Говорять, що 0 – базовий індекс.

Dim B(3,3) As Single – оголошується матриця 3×3 дійсних чисел.

Щоб задати масив, у якого 1-й елемент буде мати індекс 1, необхідно використовувати наступну конструкцію:

Dim *ІМ'Я_МАСИВУ*(1 to Розмірність) **As** *ТИП_МАСИВУ*

Наприклад: **Dim B(1 to 3, 1 to 3) As Single** – використання ключового

Dim A(1 to 12) As Integer слова **to** при оголошенні
масиву змінює базовий
індекс на 1.

Іноді розмір масиву не може бути визначений заздалегідь. У такому випадку можна оголосити масив, який називається **динамічним**, без зазначення розмірності

Dim *ІМ'Я_МАСИВУ*() **As** *ТИП_МАСИВУ*

Наприклад: **Dim Y() As Integer**

Потім додавання елементів у масив здійснюється оператором **ReDim**, який можна використовувати тільки в процедурах. Наприклад, **ReDim Y(5)**.

Елементом, які створені, необхідно присвоїти значення. Пізніше кількість елементів масиву знову можна змінити. При цьому всі значення, що зберігаються в масиві, втрачаються. Щоб зберегти їх необхідно використовувати ключове слово **Preserve**. Наприклад: **Preserve ReDim Y(7)**.

16.2. Лабораторна робота №6

Тема: «Обробка одномірних масивів»

Приклад. Дано масив X(N). Сформуванати масив Y(N) по формулі

$$y(i) = x^2(i). \text{ Знайти } S = \sum_{i=1}^N y(i).$$

Для вводу даних і запису результатів використовувати аркуш Excel.

	A	B	C	D	E	F
1	Кількість елементів			6		
2	Масив X					
3	2	3	5	-4	2	6
4	Масив Y					
5	4	9	25	16	4	36
6	Сума			94		

```
Private Sub Lab_6()
```

```
Dim X(10) As Single, Y(10) As Single
```

```
N=Cells(1,4): S=0
```

```
For i=1 to N
```

```
  x(i)=Cells(3,i)
```

```
  y(i)=x(i)^2
```

```
  S=S+y(i)
```

```
  Cells(5,i)=y(i)
```

```
Next I
```

```
Cells(6,4)=S
```

```
End Sub
```

16.3. Лабораторна робота №7

Тема: «Обробка одномірних масивів з перестановкою елементів»

Приклад. Записати елементи масиву X(N) у масив Y(N) у зворотному порядку й обчислити

$$P = \prod_{i=1}^N y(i)$$

	A	B	C	D	E	F
1	Кількість елементів			6		
2	Масив X					
3	2	3	5	-4	2	6
4	Масив Y					
5	6	2	-4	5	3	2
6	Добуток			-1440		

Private Sub Lab_7 ()

Dim X(10) As Single, Y(10) As Single

N=Cells(1,4)

P=1

For i=1 to N

 x(i)=Cells(3,i)

Next i

For i=1 to N

 y(i)=x(N-i+1): P=P*y(i)

 Cells(5,i)=y(i)

Next i

Cells(6,4)=P

End Sub

16.4. Лабораторна робота №8

Тема: «Обробка двовимірних масивів»

При виконанні лабораторних робіт №6, №7 №8 створити модуль.

Для цього виконати команду: **Вставка – Модуль.**

16.5. Створення форм. Властивості, події й методи форм.

Форма – це головний об'єкт, що утворить візуальну основу додатка. По своїй суті форма являє собою вікно, у якому можна розміщати різні керуючі елементи при створенні додатків. Для створення форм необхідно виконати команду: **Вставка – UserForm**.

У вікні Конструктора форм з'явиться форма, що має стандартний вигляд для ОС Windows.

Як і будь-який інший об'єкт VBA форма має набір властивостей (табл. 16.1). Для одержання довідки по будь-якій властивості досить виділити її у вікні властивостей і натиснути F1.



Таблиця 16.1

Основні властивості форм.

Властивість	Опис
BackColor	Кольори фона для форми.
BorderStyle	Визначає тип границі, що оточує форму.
Caption	Текст, що виводиться в заголовку форми.
Font	Визначає тип і вигляд шрифту у формі.
Height	Визначає висоту форми у твіпах.
(Name)	Ім'я об'єкта, для програми VBA.
Width	Визначає ширину форми у твіпах.

Властивості можна змінювати в режимі конструювання у вікні властивостей, або програмно в режимі виконання. Наприклад, у ході виконання програми можна змінити заголовок форми командою:

```
frmForm1.Caption="Привіт"
```

Програми в ОС Windows управляються **подіями** (табл. 16.2). Щораз, коли натискається кнопка, переміщається миша, змінюються розміри форми й тобто,

ОС генерує повідомлення, що описує виконану дію, і поміщає його в чергу повідомлень програми. Із черги повідомлення доставляється відповідному об'єкту, наприклад формі, а та генерує відповідну подію. Отже, можна скласти фрагмент програми, у якому об'єкт буде реагувати на подію певним чином, тобто будь-якій стандартній події відповідає визначена процедура. Щоб переглянути події пов'язані з формою, необхідно в режимі конструювання двічі клацнути на ній - з'явиться вікно програми, у якому клацнути на списку Процедура.

Таблиця 16.2

Основні події форм.

Подія	Опис
Initialize	Відбувається під час конфігурації й до завантаження форми на згадку.
Activate	Відбувається після завантаження форми на згадку.
Deactivate	Відбувається, якщо форма перестає бути активною.
Click	Відбувається при натисканні лівої кнопки миші на формі.

Наступний приклад змінює заголовок форми при активізації, і зменшує розмір форми після щиглика лівою кнопкою миші на формі.

```
Private Sub UserForm_Activate()
    frmForm1.Caption="Щиглик на формі зменшує її розміри"
End Sub

Private Sub UserForm_Click()
    FrmForm1.Width= FrmForm1.Width/2
    FrmFotrm1.Height= FrmForm1.Height/2
    frmForm1.Caption="Зроби це ще раз!"
End Sub
```

Також форма має набір методів і інструкцій. **Метод** визначає дію, що може бути виконане з об'єктом (табл.16.3). **Інструкція** ініціює дію. Вона може виконати метод або функцію (табл.16.4).

Основні методи форм.

Метод	Опис
Hide	Приховує об'єкт UserForm, але не вивантажує його
Show	Виводить на екран об'єкт UserForm

Таблиця 16.4

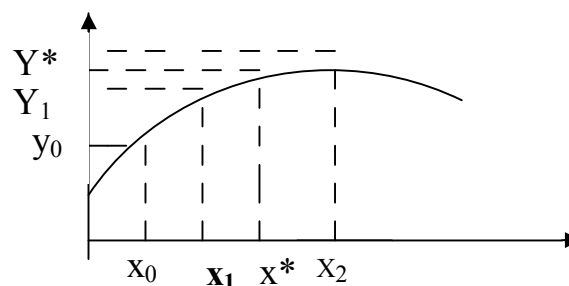
Основні інструкції форм.

Інструкція	Опис
Load	Завантажує об'єкт UserForm, але не відображає його на екрані.
Unload	Видаляє об'єкт UserForm з пам'яті.

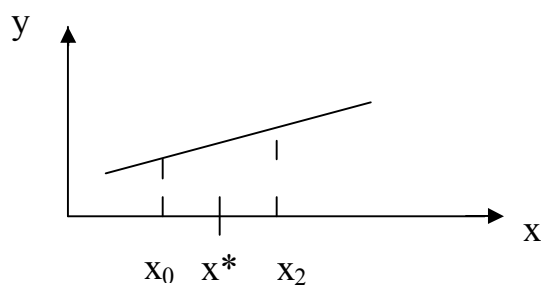
Лекція №17. МЕТОДИ ОБРОБКИ ДАНИХ. ІНТЕРПОЛЯЦІЯ

Постановка завдання. Інтерполяція: Дана таблиця значень функції y , необхідно побудувати функцію, що проходить через задані точки x , y і знайти значення цієї функції при деякому проміжному значенні x^* .

x	x_0	x_1	x_2	\dots	x_n
y	y_0	y_1	y_2	\dots	y_n



Найпростіші функції - багаточлени деякого ступеню.



$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0} \Rightarrow y = y_0 + \frac{x - x_0}{x_1 - x_0}(y_1 - y_0) \Rightarrow y = y_0 - y_0 \frac{x - x_0}{x_1 - x_0} + \frac{x - x_0}{x_1 - x_0} y_1$$

$$y = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 - \text{поліном Лагранжа 1-ого ступеня.}$$

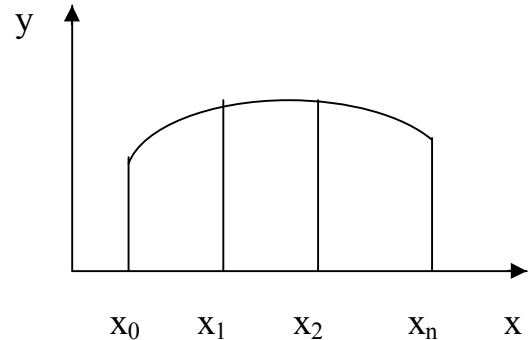
$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}, \quad l_i(x), \quad i = \overline{0,1} - \text{коефіцієнти Лагранжа.}$$

$$y = l_0(x)y_0 + l_1(x)y_1 \quad l_0(x_0) = 1, \quad l_0(x_1) = 0, \quad l_1(x_0) = 0, \quad l_1(x_1) = 1$$

$$y^* = l_0(x^*)y_0 + l_1(x^*)y_1$$

Краще інтерполяція:

1. Зменшити $x_i - x_{i-1}$
2. Підвищити ступінь полінома Лагранжа.



Багаточлен другого ступеня.

x	x_0	x_1	x_2
y	y_0	y_1	y_2

$$y = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2$$

При $n=2$ отримуємо багаточлен 2-го ступеня. При цьому треба задати три точки (тому що три точки не лежачі на одній прямій обумовлюють параболу єдиним образом).

$l_i(x), \quad i = \overline{0,2}$ - коефіцієнти Лагранжа.

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)},$$

$$l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

$$l_0(x_0) = l_1(x_1) = l_2(x_2) = 1, \quad l_0(x_1) = l_0(x_2) = 0,$$

$$l_1(x_0) = l_1(x_2) = l_2(x_0) = l_2(x_1) = 0$$

Багаточлен n-го ступеня.

$$y = P_n(x) = l_0(x)y_0 + l_1(x)y_1 + \dots + l_n(x)y_n = \sum_{j=0}^n l_j(x)y_j$$

$$l_j(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)} = \frac{\prod_{i \neq j} (x-x_i)}{\prod_{i \neq j} (x_j-x_i)}$$

$$l_j(x) = \frac{\prod_{i \neq j} (x-x_i)}{\prod_{i \neq j} (x_j-x_i)}$$

1. $l_j(x)$ - багаточлен n-го ступеня.

2. $l_j(x_j) = 1$, $l_j(x_i) = 0$, *если* $i \neq j$

При $n=5$

$$l_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)(x-x_5)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)(x_2-x_5)}$$

$$l_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)(x_0-x_5)}$$

Обчислення значень полінома Лагранжа.

Чисельник і знаменник $l_j(x)$ помножимо на $x-x_j$

$$l_j(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)(x_j-x_2)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)} = \frac{\prod_{n+1}(x)}{D_j}$$

$$\Rightarrow y = P_n(x) = \sum_{j=0}^n l_j(x) y_j = \sum_{j=0}^n \frac{\prod_{n+1}(x)}{D_j} y_j = \prod_{n+1}(x) \sum_{j=0}^n y_j / D_j$$

Таблиця обчислень значень полінома Лагранжа.

$x_i - x_j$	D_i	y_i	y_i/D_i
$x-x_0 \quad x_0-x_1 \quad x_0-x_2 \quad \dots \quad x_0-x_n$	D_0	Y_0	Y_0/D_0
$x_1-x_0 \quad x-x_1 \quad x_1-x_2 \quad \dots \quad x_1-x_n$	D_1	Y_1	Y_1/D_1
$x_2-x_0 \quad x_2-x_1 \quad x-x_2 \quad \dots \quad x_2-x_n$	D_2	Y_2	Y_2/D_2
$x_3-x_0 \quad x_3-x_1 \quad x_3-x_2 \quad x-x_3 \dots \quad x_3-x_n$	D_3	Y_3	Y_3/D_3
-----	\vdots	\vdots	\vdots
$x_n-x_0 \quad x_n-x_1 \quad x_n-x_2 \quad x_n-x_3 \quad \dots \quad x-x_n$	D_n	Y_n	Y_n/D_n

$\Pi_{n+1}(x)$ дорівнює добутку різниці на головній діагоналі:

$$\Pi_{n+1}(x) = (x-x_0)(x-x_1)(x-x_2)\dots(x-x_n),$$

D_i рівно добутку різниці в кожному рядку.

$$\Sigma = \sum_{i=0}^n \frac{y_i}{D_i}, \quad P_n(x) = \Pi_{n+1}(x) \Sigma.$$

Матриця різностей є антисиметричною (числа симетричні щодо головної діагоналі рівні по модулю й протилежні за знаком).

Приклад.

i	0	1	2
x	2	3	5
y	-2	-3	1

Ступінь полінома

Дано три значення аргументу x і відповідні ним значення y . Побудувати поліном Лагранжа й знайти його значення в проміжній точці $x^*=4$.

Лагранжа не одиницю менше числа точок, що тому інтерполює функція, яка інтерполює – квадратний тричлен ($n=2$).

$$P_2(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2$$

Складемо відповідної різниці:

$x_i - x_j$	D_i	y_i	Y_i/D_i
2 -1 -3	6	-2	-1/3
1 1 -2	-2	-3	3/2
3 2 -1	-6	1	-1/6

$$x^* - x_0 \quad x_0 - x_1 \quad x_0 - x_2$$

$$x_1 - x_0 \quad x^* - x_1 \quad x_1 - x_2$$

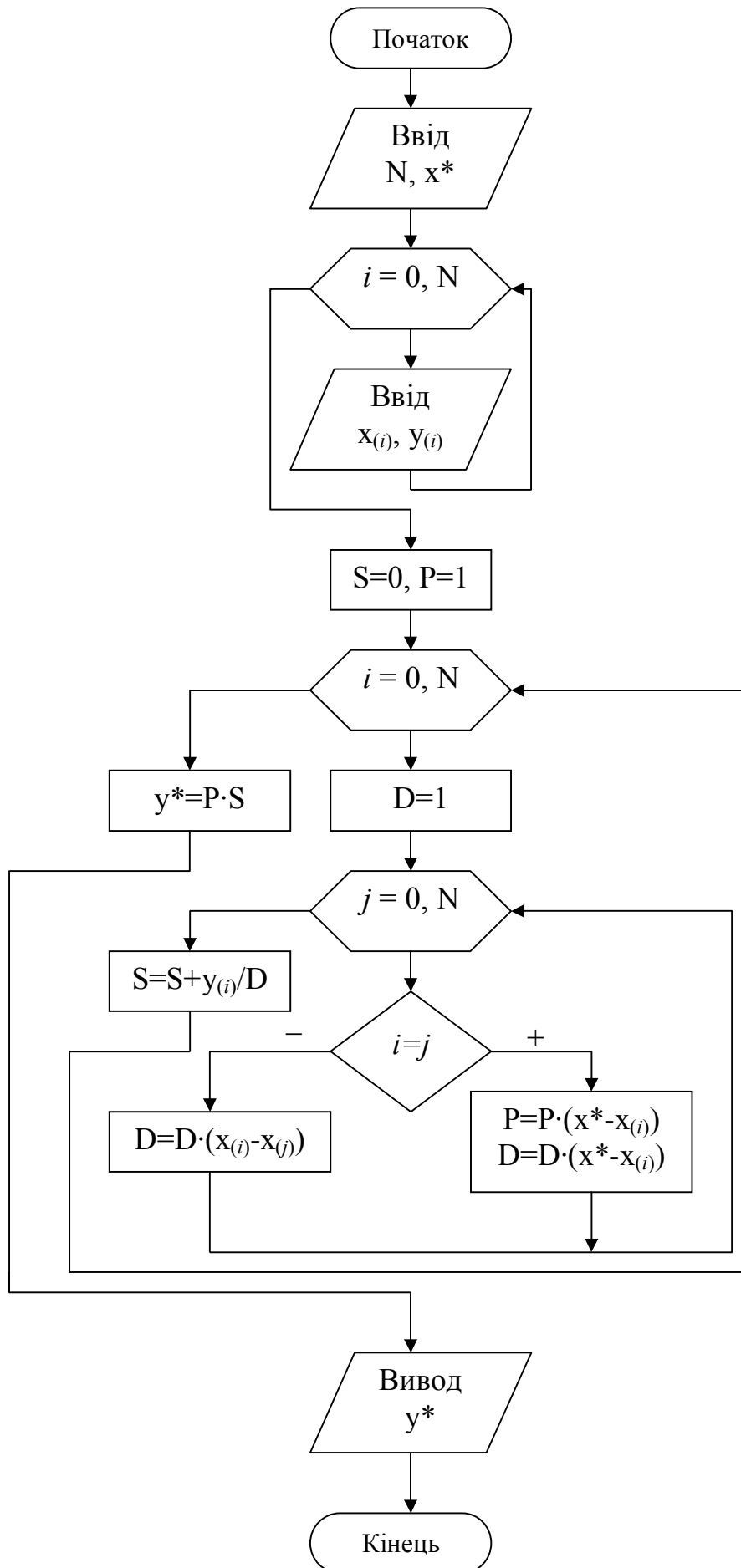
$$x_2 - x_0 \quad x_2 - x_1 \quad x^* - x_2$$

$$\Sigma = \sum_0^2 \frac{y_i}{D_i} = -\frac{1}{3} + \frac{3}{2} - \frac{1}{6} = 1, \quad \prod_3(x^*) = -2, \quad y^* = P_2(x^*) = \prod_3(x^*) \cdot \Sigma = -2$$

Якщо в останню формулу замість x^* підставити поточне значення x , то отримаємо аналітичний вид функції, що інтерполює:

$$y = x^2 - 6x + 6$$

Для автоматизації методу інтерполяції приводиться блок-схема алгоритму.



Список літератури

1. Информатика. Базовый курс. 2-е издание / Под ред. С.В. Симоновича. СПб.: Питер, 2004. – 640 с.
2. Методическое пособие. «Основы алгоритмизации»/ К.Н. Ефименко, Ю.Н. Добровольский, В.С. Ильяшенко – Донецк: ДонНТУ, 2005. – 75 с.
3. Методическое пособие. Программирование в среде Visual Basic for Application. Части 1 и 2./ В.В. Шамаев, К.Н. Ефименко и др.- Донецк: ДонНТУ, 2005. – 100 с.
4. Методичні вказівки і завдання до лабораторних робіт за курсом «Інформатика і основи програмування»/ К.М. Єфіменко, Ю.М. Добровольський. – Донецьк: ДВНЗ «ДонНТУ». – 2009. – 60 с.
5. Д. Холи, Р. Холи. Excel 2007. Трюки. – СПб.: Питер, 2008. – 368 с.
6. Б. Джелен, Т. Сирстад. Применение VBA и макросов в Microsoft Office Excel. – М.:Вильмс, 2005. – 624 с.
7. Методическое пособие. Задачи математики с элементами функционального анализа в MS Excel/ Ю.Н. Добровольский, К.Н. Ефименко, - Донецк: ДонНТУ, 2003. – 67 с.

Конспект лекцій за курсом «Інформатика і основи програмування»

Укладачі: Добровольський Юрій Миколайович
Єфіменко Костянтин Миколайович

Підп. в печать 20.06.10 р.
Різографічна печать.
Обл.-вид. а. 5,80

Формат 60x84 1/16.
Умов. печ. а. 5,70
Тираж 50 екз.

Папір KumLux.
Умов. кр.-отт. 5,75
Замовлення № 20/10

ДВНЗ «Донецький національний технічний університет»
83001, м. Донецьк, вул. Артема, 58
