

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ  
ТА СПОРТУ УКРАЇНИ  
Національний технічний університет України «Київський  
політехнічний інститут ім. І.Сікорського»

**СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ .  
ОСНОВИ АЛГОРИТМІЗАЦІЇ  
Лабораторний практикум**

для студентів ОПШ «Комп'ютерні системи та  
мережі» спеціальності 123 «Комп'ютерна  
інженерія»



Київ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ  
ТА СПОРТУ УКРАЇНИ  
Національний технічний університет України «Київський  
політехнічний інститут ім. І.Сікорського»

**СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ .  
ОСНОВИ АЛГОРИТМІЗАЦІЇ  
Лабораторний практикум**

для студентів ОПП «Комп’ютерні системи та  
мережі» спеціальності 123 «Комп’ютерна  
інженерія»

Затверджено

на засіданні кафедри системного  
програмування і спеціалізованих  
комп’ютерних систем

ФПМ НТУУ «КПІ»

Протокол № 10 від 25.05.2022

Київ НТУУ «КПІ» 2022

УДК 004.42

Структури даних та алгоритми – 1. Основи алгоритмізації: завдання до виконання лабораторних робіт з дисципліни «Структури даних та алгоритми» для студентів напрямку підготовки 123, «Комп'ютерна інженерія», ОПП «Комп'ютерні системи та мережі» [Електронне видання] / О.І.Марченко. – К.: НТУУ «КПІ», 2022. – 57 с.

Навчальне електронне видання містить завдання до виконання лабораторних робіт по кредитному модулю «Структури даних та алгоритми – 1. Основи алгоритмізації».

Наведені варіанти індивідуальних завдань. До кожної роботи надаються вказівки щодо виконання завдань, тестування програм та оформлення звіту, а також наведені контрольні питання для підготовки до захисту лабораторних робіт. Призначені для студентів напряму підготовки 123 «Комп'ютерна інженерія».

Навчальне електронне видання

Автор: *Марченко Олександр Іванович*, канд.техн.наук, доц.

Відповідальний

редактор: *Орлова М.М., кандидат техн. наук, доцент.*

Рецензент: *Заболотня Т.М., канд. техн. наук, ст. викладач*

© Марченко Олександр Іванович, 2022

За редакцією автора

## ЗМІСТ

<b>СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ – 1. ОСНОВИ АЛГОРИТМІЗАЦІЇ.....</b>	<b>1</b>
<b>ЗМІСТ .....</b>	<b>3</b>
<b>ВСТУП.....</b>	<b>5</b>
<b>1. ЛАБОРАТОРНА РОБОТА №1.1. РОЗГАЛУЖЕНІ АЛГОРИТМИ .....</b>	<b>7</b>
<b>2. ЛАБОРАТОРНА РОБОТА №1.2. АЛГОРИТМИ З ВКЛАДЕНИМИ ЦИКЛАМИ ТА МЕТОД ДИНАМІЧНОГО ПРОГРАМУВАННЯ .....</b>	<b>14</b>
<b>3. ЛАБОРАТОРНА РОБОТА №1.3. ВИКОРИСТАННЯ КЕРУЮЧИХ КОНСТРУКЦІЙ РОЗГАЛУЖЕННЯ ТА ЦИКЛІВ ПРИ РОБОТІ З ОДНОВИМІРНИМИ МАСИВАМИ .....</b>	<b>19</b>
<b>4. ЛАБОРАТОРНА РОБОТА №1.4. АЛГОРИТМИ ПЕРЕТВОРЕННЯ ОДНОВИМІРНИХ МАСИВІВ (ВЕКТОРІВ) .....</b>	<b>24</b>
<b>5. ЛАБОРАТОРНА РОБОТА №1.5. АЛГОРИТМИ ЛІНІЙНОГО ПОШУКУ.....</b>	<b>32</b>

**6. ЛАБОРАТОРНА РОБОТА №1.6. АЛГОРИТМИ  
ОБХОДУ ДВОВИМІРНИХ МАСИВІВ (МАТРИЦЬ) ..... 40**

## ВСТУП

Дисципліна "Структури даних та алгоритми" є базовою спеціальною нормативною дисципліною підготовки фахівців рівня бакалавр з напрямку 050102 «Комп'ютерна інженерія», а також студентів спеціальностей 8.091.501 "Комп'ютерні системи та мережі", 8.091.502 "Системне програмування" та 8.091.503 "Спеціалізовані комп'ютерні системи" і викладається у 1-му та 2-му семестрах.

Вона повинна передувати всім програмістським дисциплінам крім початкового курсу програмування, з яким викладається одночасно.

Кредитний модуль «Структури даних та алгоритми – 1. Основи алгоритмізації» призначений для вивчення принципів та концепцій, які покладені в основу розробки обчислювальних і керуючих алгоритмів та відповідних їм структур даних, а також структурної та об'єктно-орієнтованої методології програмування.

Цикл лабораторних робіт складається з шести робіт і призначений для покриття практичної частини першого кредитного модуля дисципліни «Структури даних та алгоритми».

Роботи дозволяють отримати практичні навички складання розгалужених та циклічних алгоритмів, а також використання базових структур даних, що використовуються в програмуванні.

Завдання до кожної роботи включають:

- постановку завдання та вимоги до алгоритмів та програм, що повинні розробити студенти;
- вказівки до оформлення звіту та тестування розробленого алгоритму і відповідної йому програми;
- контрольні питання для самоконтролю та підготовки до захисту лабораторної роботи;
- варіанти індивідуальних завдань.

# 1. ЛАБОРАТОРНА РОБОТА №1.1.

## РОЗГАЛУЖЕНІ АЛГОРИТМИ

### Мета лабораторної роботи

Метою лабораторної роботи №1.1. «Розгалужені алгоритми» є засвоєння теоретичного матеріалу та набуття практичних навичок використання керуючих конструкцій розгалуження та булевих (логічних) операцій.

### Постановка задачі

Задано дійсне число  $x$ . Визначити значення заданої за варіантом кусочно-безперервної функції  $y(x)$ , якщо воно існує, або вивести на екран повідомлення про неіснування функції для заданого  $x$ .

Розв'язати задачу двома способами (написати дві програми):

1) в програмі дозволяється використовувати тільки одиничні операції порівняння ( $=$ ,  $<$ ,  $>$ ,  $<=$ ,  $>=$ ) і **не** дозволяється використовувати булеві (логічні) операції (**not**, **and**, **or**, тощо);

2) в програмі необхідно обов'язково використати булеві (логічні) операції (**not**, **and**, **or**, тощо); використання булевих операцій не повинно бути надлишковим.



## Зміст звіту

1. Загальна постановка задачі та завдання для конкретного варіанту.
2. Текст обох програм.
3. Тестування обох програм. При тестуванні кожної з програм необхідно підібрати такий набір вхідних значень  $x$ , щоб коректність всіх гілок алгоритмів була протестована.
4. В якості результату роздрукувати дані тестування для всіх вхідних значень з тестового набору для кожної з програм окремо.

## Контрольні питання

1. Класифікація керуючих конструкцій розгалуження.
2. Запис керуючих конструкцій розгалуження діаграмами дій.
3. Запис керуючих конструкцій розгалуження мовою програмування.
4. Таблиця істинності булевої (логічної) операції *not*.
5. Таблиця істинності булевої (логічної) операції *and*.
6. Таблиця істинності булевої (логічної) операції *or*.
7. Запис складних булевих (логічних) виразів.
8. Які існують варіанти використання вкладених умовних конструкцій (при наявності альтернативної гілки та при її відсутності) та особливості запису таких вкладених конструкцій мовою програмування?

## Варіанти індивідуальних завдань

### Варіант 1

$$y = \begin{cases} 10x^3 + 7x/5 + 2 & , x \in [-49, -10] \cup (0, 10] \\ -x + 9 & , x \in (20, +\infty) \end{cases}$$

### Варіант 2

$$y = \begin{cases} x^3 - 5x^2 & , x \in (0, 5] \\ x^2 - 3 & , x \in [-32, -20] \cup (10, +\infty) \end{cases}$$

### Варіант 3

$$y = \begin{cases} -14x - 20 & , x \in (-21, 3] \cup (12, +\infty) \\ 13x^2/11 - 6 & , x \in (-\infty, -41] \end{cases}$$

### Варіант 4

$$y = \begin{cases} 4x^2 + 2 & , x \in (-15, 3] \\ 3x^3/4 - 5 & , x \in (-\infty, -30] \cup (20, +\infty) \end{cases}$$

### Варіант 5

$$y = \begin{cases} x^3 - 6 & , x \in (-10, -5] \cup (5, 15] \\ 2x^3 - 3x + 2 & , x \in [25, +\infty) \end{cases}$$

### Варіант 6

$$y = \begin{cases} x^3 + 14 & , x \in [2, 7] \\ -4x^3 + 3x - 7 & , x \in (-13, -3] \cup (14, +\infty) \end{cases}$$

**Варіант 7**

$$y = \begin{cases} -3x^2/5 + 9 & , x \in [0,8] \cup [16, +\infty) \\ 15x - 2 & , x \in (-\infty, -1) \end{cases}$$

**Варіант 8**

$$y = \begin{cases} -6x^2 + 8 & , x \in [0,7) \\ -x^3/7 + 10 & , x \in (-\infty, -10] \cup [11, +\infty) \end{cases}$$

**Варіант 9**

$$y = \begin{cases} -9x^3 + 5x^2 & , x \in (2,12] \cup (22,32) \\ -x^2 - 12 & , x \in (-\infty, 0] \end{cases}$$

**Варіант 10**

$$y = \begin{cases} -5x^3 + 10 & , x \in [8,23) \\ 2x^3 + 8x^2 & , x \in (-\infty, -19) \cup (-3,0] \end{cases}$$

**Варіант 11**

$$y = \begin{cases} 10x^3 + 7x/5 + 2 & , x \in [-49, -10) \cup (0,10] \\ -x + 9 & , x \in (20, +\infty) \end{cases}$$

**Варіант 12**

$$y = \begin{cases} x^3 - 5x^2 & , x \in (0,5) \\ x^2 - 3 & , x \in [-32, -20) \cup (10, +\infty) \end{cases}$$

**Варіант 13**

$$y = \begin{cases} -14x - 20 & , x \in (-21,3] \cup (12, +\infty) \\ 13x^2/11 - 6 & , x \in (-\infty, -41] \end{cases}$$

**Варіант 14**

$$y = \begin{cases} 4x^2 + 2 & , x \in (-15,3] \\ 3x^3/4 - 5 & , x \in (-\infty, -30] \cup (20, +\infty) \end{cases}$$

**Варіант 15**

$$y = \begin{cases} x^3 - 6 & , x \in (-10,-5] \cup (5,15] \\ 2x^3 - 3x + 2 & , x \in [25, +\infty) \end{cases}$$

**Варіант 16**

$$y = \begin{cases} x^3 + 14 & , x \in [2,7] \\ -4x^3 + 3x - 7 & , x \in (-13,-3] \cup (14, +\infty) \end{cases}$$

**Варіант 17**

$$y = \begin{cases} -3x^2/5 + 9 & , x \in [0,8] \cup [16, +\infty) \\ 15x - 2 & , x \in (-\infty, -1) \end{cases}$$

**Варіант 18**

$$y = \begin{cases} -6x^2 + 8 & , x \in [0,7) \\ -x^3/7 + 10 & , x \in (-\infty, -10] \cup [11, +\infty) \end{cases}$$

**Варіант 19**

$$y = \begin{cases} -9x^3 + 5x^2 & , x \in (2,12] \cup (22,32) \\ -x^2 - 12 & , x \in (-\infty, 0] \end{cases}$$

**Варіант 20**

$$y = \begin{cases} -5x^3 + 10 & , x \in [8,23) \\ 2x^3 + 8x^2 & , x \in (-\infty, -19) \cup (-3,0] \end{cases}$$

**Варіант 21**

$$y = \begin{cases} 10x^3 + 7x/5 + 2 & , x \in [-49, -10) \cup (0,10] \\ -x + 9 & , x \in (20, +\infty) \end{cases}$$

**Варіант 22**

$$y = \begin{cases} x^3 - 5x^2 & , x \in (0,5] \\ x^2 - 3 & , x \in [-32, -20) \cup (10, +\infty) \end{cases}$$

**Варіант 23**

$$y = \begin{cases} -14x - 20 & , x \in (-21,3] \cup (12, +\infty) \\ 13x^2/11 - 6 & , x \in (-\infty, -41] \end{cases}$$

**Варіант 24**

$$y = \begin{cases} 4x^2 + 2 & , x \in (-15,3] \\ 3x^3/4 - 5 & , x \in (-\infty, -30] \cup (20, +\infty) \end{cases}$$

**Варіант 25**

$$y = \begin{cases} x^3 - 6 & , x \in (-10, -5] \cup (5, 15] \\ 2x^3 - 3x + 2 & , x \in [25, +\infty) \end{cases}$$

**Варіант 26**

$$y = \begin{cases} x^3 + 14 & , x \in [2, 7] \\ -4x^3 + 3x - 7 & , x \in (-13, -3] \cup (14, +\infty) \end{cases}$$

**Варіант 27**

$$y = \begin{cases} -3x^2/5 + 9 & , x \in [0, 8] \cup [16, +\infty) \\ 15x - 2 & , x \in (-\infty, -1) \end{cases}$$

**Варіант 28**

$$y = \begin{cases} -6x^2 + 8 & , x \in [0, 7] \\ -x^3/7 + 10 & , x \in (-\infty, -10] \cup [11, +\infty) \end{cases}$$

**Варіант 29**

$$y = \begin{cases} -9x^3 + 5x^2 & , x \in (2, 12] \cup (22, 32) \\ -x^2 - 12 & , x \in (-\infty, 0] \end{cases}$$

**Варіант 30**

$$y = \begin{cases} -5x^3 + 10 & , x \in [8, 23) \\ 2x^3 + 8x^2 & , x \in (-\infty, -19) \cup (-3, 0] \end{cases}$$

## **2. ЛАБОРАТОРНА РОБОТА №1.2.**

### **АЛГОРИТМИ З ВКЛАДЕНИМИ ЦИКЛАМИ ТА МЕТОД ДИНАМІЧНОГО ПРОГРАМУВАННЯ**

#### **Мета лабораторної роботи**

Метою лабораторної роботи №1.2. «Алгоритми з вкладеними циклами та метод динамічного програмування» є засвоєння циклами та метод динамічного програмування» є засвоєння теоретичного матеріалу та набуття практичних навичок використання різних циклічних керуючих конструкцій, вкладених циклів, методу динамічного програмування та обчислення кількості операцій алгоритмів.

#### **Постановка задачі**

1. Задане натуральне число  $n$ . Вирахувати значення заданої формули за варіантом.
2. Для вирішення задачі написати дві програми:
  - 1) перша програма повинна використовувати для обчислення формули вкладені цикли;
  - 2) друга програма повинна виконати обчислення формули за допомогою одного циклу з використанням методу динамічного програмування.
3. Виконати розрахунок кількості операцій для кожного з алгоритмів за методикою, викладеною на лекції, додавши до неї підрахунок кількості викликів стандартних функцій.

4. Програма має правильно вирішувати поставлену задачу при будь-якому заданому  $n$ , для якого результат обчислення може бути коректно представлений типом *double*.

5. Результуючі дані вивести у форматі з сімома знаками після крапки.

### Зміст звіту

1. Загальна постановка задачі та завдання для конкретного варіанту.

2. Текст обох програм.

3. Розрахунки кількості операцій для кожного з алгоритмів згідно методики, викладеної на лекціях.

4. Тестування програм. З метою створення еталону для перевірки коректності програм потрібно розписати у звіті обчислення заданої формули для  $n=3$  з підстановкою фактичних чисел і виконати обчислення цієї формули на калькуляторі.

5. В якості результату тестування роздрукувати результати виконання обох програм для  $n=1$ ,  $n=2$  та  $n=3$ .

6. Результати обчислень обома програмами та за допомогою калькулятора повинні співпадати.

7. Результати розрахунків обома програмами для заданого значення  $n$ .



## **Контрольні питання**

1. Класифікація циклічних керуючих конструкцій алгоритмів та особливості їх використання.
2. Запис циклічних керуючих конструкцій алгоритмів діаграмами дій.
3. Запис циклічних керуючих конструкцій алгоритмів мовою програмування.
4. Методика спрощеного підрахунку кількості операцій алгоритмів.
5. Метод динамічного програмування: визначення та значимість при створенні алгоритмів.

## Варіанти завдань

1.	$S = \sum_{i=1}^n \frac{4^i - i}{\prod_{j=1}^i (j + \cos(j))}$	2.	$P = \prod_{i=1}^n \frac{i^2 + 1}{\sum_{j=1}^i (j + \sin(j))}$
3.	$S = \sum_{i=1}^n \frac{3^i \cdot \ln(i+1)}{\prod_{j=1}^i (2j+1)}$	4.	$P = \prod_{i=1}^n \frac{2i \cdot \ln(i+3)}{\sum_{j=1}^i (2j+1)}$
5.	$S = \sum_{i=1}^n \frac{i \cdot 2^i}{\prod_{j=1}^i \left( \frac{j+1}{10} \right)}$	6.	$P = \prod_{i=1}^n \frac{i + \sqrt{i}}{\sum_{j=1}^i \left( \frac{4}{3} j - 1 \right)}$
7.	$S = \sum_{i=1}^n \frac{i \cdot (i+1)}{\prod_{j=1}^i ((j+1) \cdot \sin(j))}$	8.	$P = \prod_{i=1}^n \frac{i \cdot \sqrt{i}}{\sum_{j=1}^i (j \cdot \sin(j) + j)}$
9.	$S = \sum_{i=1}^n \frac{(2^i + 1)^2}{\prod_{j=1}^i (j+1)}$	10.	$P = \prod_{i=1}^n \frac{i \cdot (i+1)}{\sum_{j=1}^i (\sin(j) + 1)}$
11.	$S = \sum_{i=1}^n \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))}$	12.	$S = \sum_{i=1}^n \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))}$
13.	$S = \sum_{i=1}^n \frac{3 - \sin^2(i)}{\prod_{j=1}^i (\ln(j+2))}$	14.	$P = \prod_{i=1}^n \frac{\cos(i) + 1}{\sum_{j=1}^i \sin(j)}$
15.	$S = \sum_{i=1}^n \frac{\sin(i) + 2}{i + \prod_{j=1}^i \sin(j)}$	16.	$P = \prod_{i=1}^n \frac{3 - \cos^2(i)}{\sum_{j=1}^i \ln(j+2)}$

17.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (j + \cos(j))}{4^i - i}$	$\frac{1}{8}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (j + \sin(j))}{i^2 + 1}$
19.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (2j + 1)}{3i \cdot \ln(i + 1)}$	$\frac{2}{0}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (2j + 1)}{2i \cdot \ln(i + 3)}$
21.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i \left( \frac{j+2}{10} \right)}{i \cdot 2^i}$	$\frac{2}{2}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i \left( \frac{4}{3} j - 1 \right)}{i + \sqrt{i}}$
23.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$	$\frac{2}{4}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (j \cdot \sin(j) + j)}{i \cdot \sqrt{i}}$
25.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (j+1)}{(2^i + 1)^2}$	$\frac{2}{6}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (\sin(j) + 1)}{i \cdot (i+1)}$
27.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (j - \cos(j))}{i + \cos(i)}$	$\frac{2}{8}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (j - \cos(j))}{i + \sin(i)}$
29.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (\ln(j+2))}{3 - \sin^2(i)}$	$\frac{3}{0}$ ·	$P = \prod_{i=1}^n \frac{\sum_{j=1}^i \sin(j)}{\cos(i) + 1}$

### 3. ЛАБОРАТОРНА РОБОТА №1.3.

## ВИКОРИСТАННЯ КЕРУЮЧИХ КОНСТРУКЦІЙ РОЗГАЛУЖЕННЯ ТА ЦИКЛІВ ПРИ РОБОТІ З ОДНОВИМІРНИМИ МАСИВАМИ

### Мета лабораторної роботи

Метою лабораторної роботи №1.3. «Використання керуючих конструкцій розгалуження та циклів при роботі з одновимірними масивами» є засвоєння теоретичного матеріалу та набуття практичних навичок створення складних комбінованих алгоритмів з використанням конструкцій розгалуження, циклічних керуючих конструкцій та одновимірних масивів.

### Постановка задачі

1. Задано дійсні числа  $a, b$ , ціле число  $n$  та одновимірний масив (вектор) дійсних випадкових чисел  $Y[n]$ .
2. Отримати одновимірний масив (вектор) дійсних чисел  $Z[n]$ , елементи якого  $z_i = f(y_1, y_2, \dots, y_n)$ , де  $f$  – задана за варіантом функція, а  $y_i$  – елементи вектора  $Y[n]$ .
3. Отримати результуюче значення  $R = g(z_1, z_2, \dots, z_n)$ , де  $g$  – задана за варіантом функція, а  $z_i$  – елементи вектора  $Z[n]$ .
4. Значення змінних  $a, b$  та  $n$  є початковими даними, які вибираються самостійно так, щоб функція  $f$  існувала при даних значеннях цих змінних.

5. Номери функцій  $f$  та  $g$  визначити за варіантом завдання (див. таблицю 1).

6. Програма має правильно вирішувати поставлену задачу при вхідних даних  $a, b, n$ .

7. Значення початкового масиву  $Y[n]$ , проміжного масиву  $Z[n]$  та результуюче значення  $R$  вивести у форматі з трьома знаками після крапки.

### Зміст звіту

1. Загальна постановка задачі та завдання для конкретного варіанту.

2. Текст програми, вхідні дані.

3. Тестові початкові дані для налагодження програми і результати виконання програми для цих тестових даних.

4. Діапазон генерування випадкових чисел для формування початкового масиву  $Y[n]$  повинен бути встановлений таким чином, щоб згенеровані елементи масиву належали різним діапазонам існування заданих функцій.

5. Результати розрахунків для довільних значень  $a, b$  та  $n$ .

### Контрольні питання

1. Класифікація керуючих конструкцій та особливості їх використання.

2. Класифікація циклічних керуючих конструкцій алгоритмів та особливості їх використання.

3. Які існують варіанти використання вкладених умовних конструкцій (при наявності альтернативної гілки та при її відсутності) та особливості запису таких вкладених конструкцій мовою програмування?

4. Яким умовам повинен задовольняти лічильник (параметр) конструкції циклу з лічильником та початкове і кінцеве значення лічильника?

5. Чи відповідає структурному стилю програмування використання значення лічильника (параметра) конструкції циклу з лічильником після завершення циклу та після переривання роботи циклу конструкцією виходу з циклу?

6. Які подібності та відмінності мають циклічні конструкції з передумовою та післяумовою?

### Варіанти завдань

*Таблиця 1*

№ варіанту	Функція f	Функція g	№ варіанту	Функція f	Функція g
<b>1</b>	1	1	<b>16</b>	6	9
<b>2</b>	2	2	<b>17</b>	7	10
<b>3</b>	3	3	<b>18</b>	8	1
<b>4</b>	4	4	<b>19</b>	9	2
<b>5</b>	5	5	<b>20</b>	10	3
<b>6</b>	6	6	<b>21</b>	1	7
<b>7</b>	7	7	<b>22</b>	2	8
<b>8</b>	8	8	<b>23</b>	3	9
<b>9</b>	9	9	<b>24</b>	4	10
<b>10</b>	10	10	<b>25</b>	5	1

<b>11</b>	1	4	<b>26</b>	6	2
<b>12</b>	2	5	<b>27</b>	7	3
<b>13</b>	3	6	<b>28</b>	8	4
<b>14</b>	4	7	<b>29</b>	9	5
<b>15</b>	5	8	<b>30</b>	10	6

**Функція  $f(y_1, \dots, y_n)$  для отримання ряду значень  $z_i$**

1.  $z_i = f(y_1, \dots, y_n) = \begin{cases} y_i + 5b \text{ нпу } y_i \notin [11, 33], \\ y_i - 10a \text{ нпу } y_i \in [11, 33]. \end{cases}$
2.  $z_i = f(y_1, \dots, y_n) = \begin{cases} ay_i^2 \text{ нпу } |y_i| \leq 2, \\ 0,5by_i \text{ нпу } |y_i| > 2. \end{cases}$
3.  $z_i = f(y_1, \dots, y_n) = \begin{cases} 2a + y_i^2 \text{ нпу } y_i \in (0, 15), \\ 2b \text{ нпу } y_i \notin (0, 15). \end{cases}$
4.  $z_i = f(y_1, \dots, y_n) = \begin{cases} 5 - y_i \text{ нпу } |y_i| > a, \\ 3by_i \text{ нпу } |y_i| \leq a, \\ 2y_i^3 \text{ нпу } |y_i| \leq 4b, \end{cases}$
5.  $z_i = f(y_1, \dots, y_n) = \begin{cases} i & i \\ ay_i - 2, \text{ нпу } |y_i| > 4b. \end{cases}$
6.  $z_i = f(y_1, \dots, y_n) = \begin{cases} 7 - by_i \text{ нпу } y_i \notin (-20, 10), \\ 1 - ay_i \text{ нпу } y_i \in (-20, 10). \end{cases}$
7.  $z_i = f(y_1, \dots, y_n) = \begin{cases} y_i + ab \text{ нпу } y_i \in [-15, 25], \\ \frac{2a}{i} + 3b \text{ нпу } y_i \notin [-15, 25], \\ \text{нпу } |y_i| > 7a, \end{cases}$
8.  $z_i = f(y_1, \dots, y_n) = \begin{cases} |y_i| + 3 & i \\ 6by_i \text{ нпу } |y_i| \leq 7a. \end{cases}$
9.  $z_i = f(y_1, \dots, y_n) = \begin{cases} 10a - y_i \text{ нпу } y_i \in (-50, 0), \\ 1 + y_i^2 \text{ нпу } y_i \notin (-50, 0), \\ ay_i^3 \text{ нпу } |y_i| \leq 10, \end{cases}$
10.  $z_i = f(y_1, \dots, y_n) = \begin{cases} i & i \\ 1 + y_i \text{ нпу } |y_i| > 10. \end{cases}$

**Функція  $g(z_1, \dots, z_n)$  для отримання результуючого значення  $R$**

1.  $R = g(z_1, \dots, z_n) = \min_{1 \leq i \leq n} ((-1)^i a z_i)$
2.  $R = g(z_1, \dots, z_n) = \max_{1 \leq i \leq n} ((-1)^i b + z_i)$
3.  $R = g(z_1, \dots, z_n) = \prod_{i=1}^n (-1)^i \frac{z_i + a}{ib}$
4.  $R = g(z_1, \dots, z_n) = \max_{1 \leq i \leq n} |z_i - (-1)^i ab|$
5.  $R = g(z_1, \dots, z_n) = \min_{1 \leq i \leq n} |(-1)^i b z_i - 4ia|$
6.  $R = g(z_1, \dots, z_n) = \sum_{i=1}^n (-1)^{i+1} (a z_i - 3b)$
7.  $R = g(z_1, \dots, z_n) = \sum_{i=1}^n (-1)^i \left( \sqrt{\frac{z_i^2 + a^2}{z_i}} - z_i \right)^2$
8.  $R = g(z_1, \dots, z_n) = \prod_{i=1}^n (-1)^{i+1} \frac{a + z_i^2}{2 + b^2}$
9.  $R = g(z_1, \dots, z_n) = \sum_{i=1}^n (-1)^i \sqrt{10b + a z_i^2}$
10.  $R = g(z_1, \dots, z_n) = \prod_{i=1}^n (-1)^{i+1} (6b z_i - i^2 a)$



## 4. ЛАБОРАТОРНА РОБОТА №1.4.

### АЛГОРИТМИ ПЕРЕТВОРЕННЯ ОДНОВИМІРНИХ МАСИВІВ (ВЕКТОРІВ)

#### Мета лабораторної роботи

Метою лабораторної роботи №1.4. «Алгоритми перетворення одновимірних масивів (векторів)» є засвоєння теоретичного матеріалу та набуття практичних навичок створення складних алгоритмів перетворення даних у одновимірних масивах.

#### Постановка задачі

Задано натуральне число  $n$  і одновимірний масив цілих чисел  $A[4n]$ , який складається з чотирьох частин по  $n$  елементів кожна.

1	n	n+1	2n	2n+1	3n	3n+1	4n
1-ша частина	2-га частина	3-тя частина	4-та частина				

Перетворити масив «на тому ж місці» (тобто результат перетворення повинен бути у тому ж масиві  $A$ ), виконавши перестановки елементів масиву  $A$  у порядку, заданому за варіантом.

Розв'язати задачу двома способами (написати дві програми):

1) для виконання перетворення масиву  $A$  в програмі дозволяється використовувати один додатковий проміжний масив  $B$  розміром не більше  $4n$  елементів. Кращим варіантом рішення буде використання проміжного масиву  $B$  розміром у  $n$  елементів;

2) для виконання перетворення масиву  $A$  в програмі дозволяється використати тільки одну просту додаткову проміжну змінну цілого типу.

### **Зміст звіту**

1. Загальна постановка задачі та завдання для конкретного варіанту.

2. Текст обох програм.

3. Тестування обох програм. При тестуванні кожної з програм треба підібрати такий набір вхідних значень масиву  $A$ , щоб можна було легко відстежити коректність перетворення масиву.

Необхідно протестувати мінімум три випадки: 1)  $n = 1$ ; 2)  $n$  – парне; 3)  $n$  – непарне.

4. В якості результату роздрукувати дані тестування для всіх вхідних значень масиву  $A$  з тестового набору для кожної з програм окремо.

### **Контрольні питання**

1. Класифікація масивів.

2. Запис оголошення та використання одновимірних масивів (векторів) діаграмами дій.

3. Запис оголошення та використання одновимірних масивів (векторів) мовою програмування.

4. Що означає термін «перетворення масиву»?

5. Що є необхідним для виконання перетворення масиву?

## Варіанти індивідуальних завдань

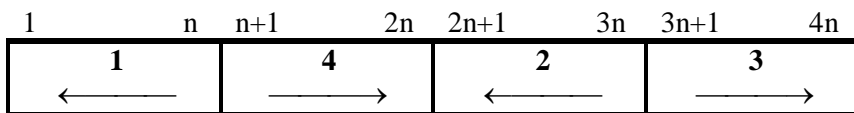
### Умовні позначення.

1. Цифри 1, 2, 3, 4 позначають номер частини початкового масиву  $A$ . У варіантах завдання ці цифри стоять на тих місцях, на які повинна бути переставлена позначена вказаним номером частина масиву  $A$ .

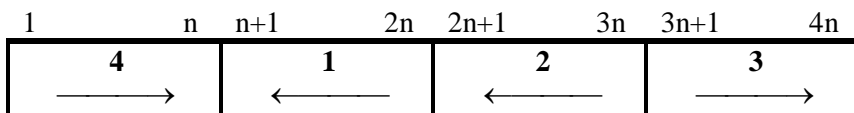
2. Стрілка «вправо» ( $\longrightarrow$ ) позначає, що порядок взаємного розташування елементів відповідної частини масиву  $A$  після перестановки повинен бути збережений таким самим, як у початковому масиві.

3. Стрілка «вліво» ( $\longleftarrow$ ) позначає, що порядок взаємного розташування елементів відповідної частини масиву  $A$  після перестановки повинен бути оберненим до того розташування, яке було у початковому масиві, тобто під час перестановки цієї частини масиву виконується «дзеркальне відображення» її елементів.

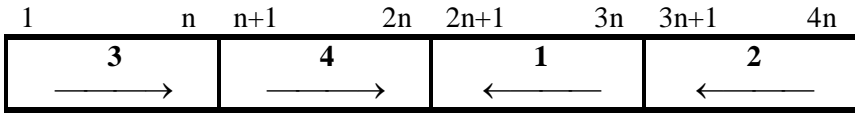
### Варіант 1



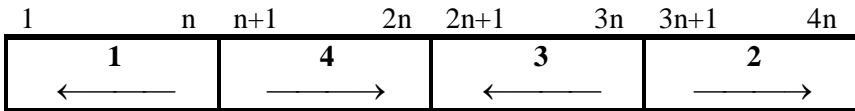
### Варіант 2



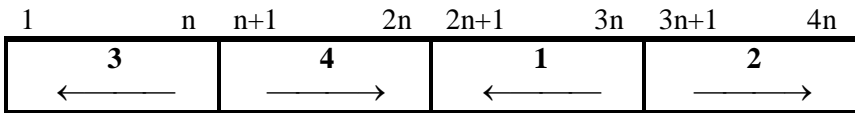
**Варіант 3**



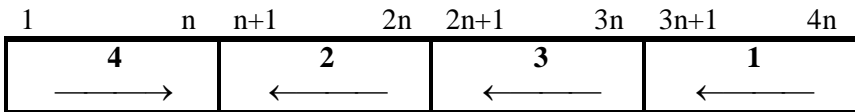
**Варіант 4**



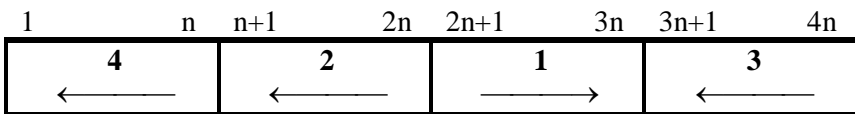
**Варіант 5**



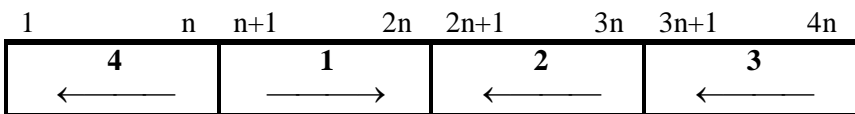
**Варіант 6**



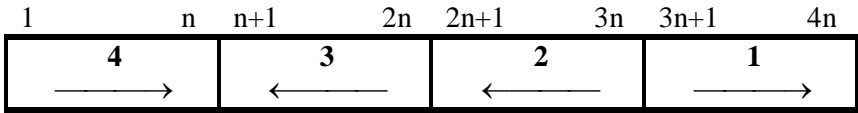
**Варіант 7**



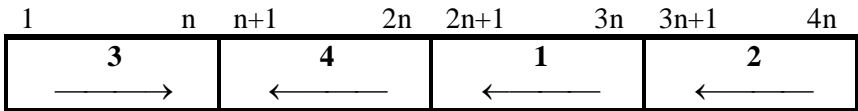
**Варіант 8**



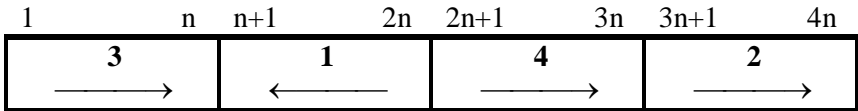
**Варіант 9**



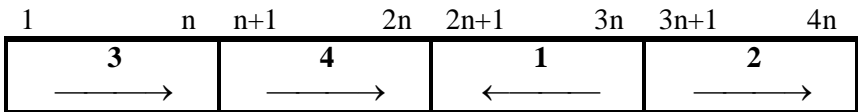
**Варіант 10**



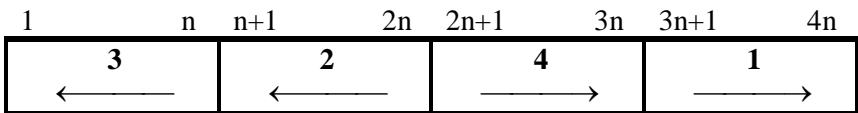
**Варіант 11**



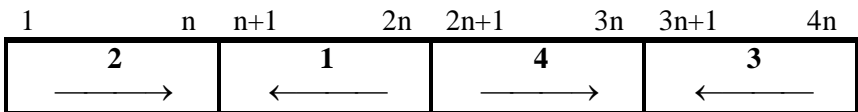
**Варіант 12**

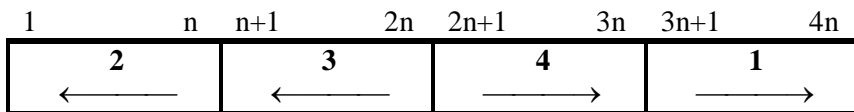
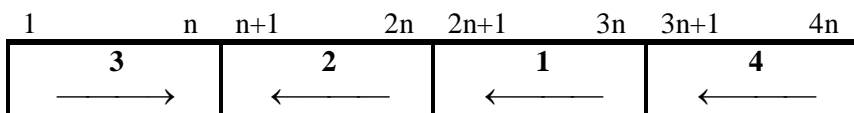
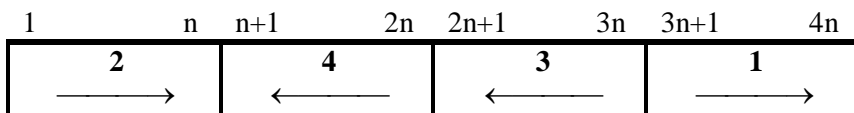
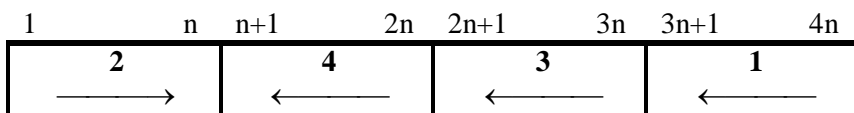
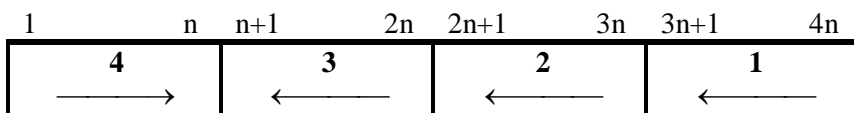
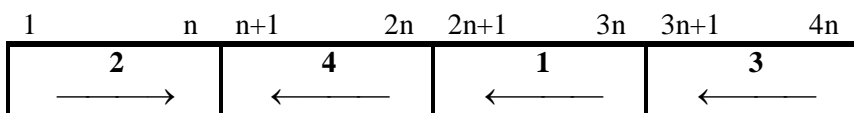


**Варіант 13**

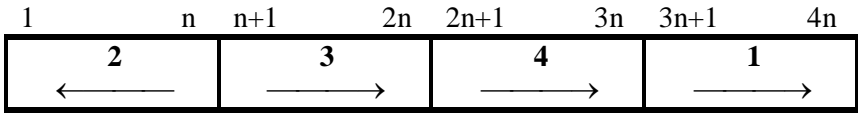


**Варіант 14**

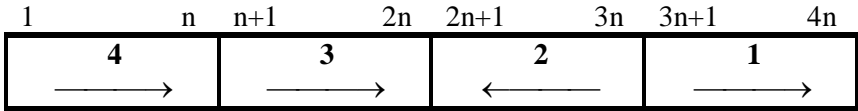


**Варіант 15****Варіант 16****Варіант 17****Варіант 18****Варіант 19****Варіант 20**

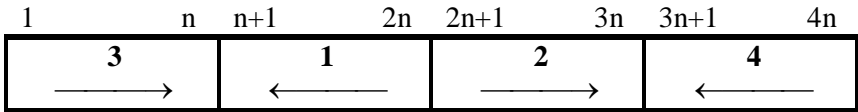
**Варіант 21**



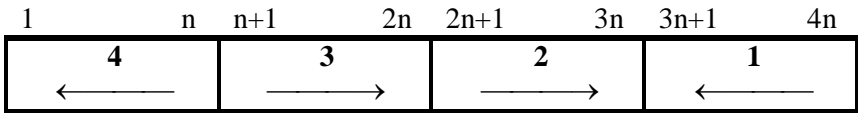
**Варіант 22**



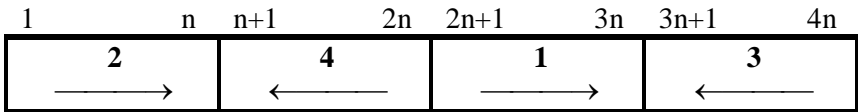
**Варіант 23**



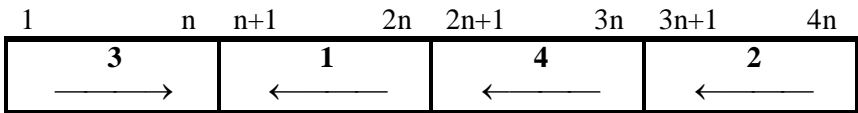
**Варіант 24**



**Варіант 25**



**Варіант 26**



**Варіант 27**

1	n	n+1	2n	2n+1	3n	3n+1	4n
3 ←	4 →	2 →	1 →				

**Варіант 28**

1	n	n+1	2n	2n+1	3n	3n+1	4n
4 →	3 ←	1 ←	2 ←				

**Варіант 29**

1	n	n+1	2n	2n+1	3n	3n+1	4n
3 ←	1 →	2 →	3 →				

**Варіант 30**

1	n	n+1	2n	2n+1	3n	3n+1	4n
2 ←	1 →	4 →	3 ←				



## 5. ЛАБОРАТОРНА РОБОТА №1.5. АЛГОРИТМИ ЛІНІЙНОГО ПОШУКУ

### Мета лабораторної роботи

Метою лабораторної роботи №1.5. «Алгоритми лінійного пошуку» є засвоєння теоретичного матеріалу та набуття практичних навичок рішення задачі пошуку заданої категорії елементів за допомогою різних алгоритмів методу лінійного пошуку у двовимірних масивах.

### Постановка задачі

1. Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.
2. Розміри матриці  $m$  та  $n$  взяти самостійно у межах від 7 до 10.
3. Виконати тестування та налагодження програми на комп'ютері. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

### Зміст звіту

1. Загальна постановка задачі та завдання для конкретного варіанту.

2. Текст програми, вхідні дані.
3. Тести для налагодження програми і результати, отримані для них на комп'ютері.

### **Контрольні питання**

1. Які умові закінчення лінійного пошуку?
2. Написати алгоритм лінійного пошуку за допомогою циклу з лічильником та конструкцією виходу з циклу.
3. Написати алгоритм лінійного пошуку за допомогою циклу з передумовою і без використання «бар'єру».
4. Написати алгоритм лінійного пошуку за допомогою циклу з передумовою і з використанням «бар'єру».
5. Чим відрізняється загальна поведінка алгоритмів лінійного пошуку з використанням «бар'єру» і без його використання?

### **Варіанти індивідуальних завдань**

*Розміри матриці  $m$  і  $n$  взяти самостійно у межах від 7 до 10.*

#### **Варіант 1**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному рядку матриці визначити присутність заданого дійсного числа  $X$  і його місцезнаходження (координати).

## **Варіант 2**

Задано матрицю дійсних чисел  $A[n,n]$ . У головній діагоналі матриці знайти перший додатний і останній від'ємний елементи, а також поміняти їх місцями.

## **Варіант 3**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках знайти в ній перший додатний елемент і його місцезнаходження (координати).

## **Варіант 4**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному стовпчику матриці визначити присутність заданого дійсного числа  $X$  і його місцезнаходження (координати).

## **Варіант 5**

Задано матрицю дійсних чисел  $A[n,n]$ . У побочній діагоналі матриці знайти перший мінімальний і останній максимальний елементи, а також поміняти їх місцями.

## **Варіант 6**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по стовпчиках знайти в ній перший мінімальний елемент і його місцезнаходження (координати).

### **Варіант 7**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному рядку матриці знайти останній від'ємний елемент і його місцезнаходження (координати).

### **Варіант 8**

Задано матрицю дійсних чисел  $A[n,n]$ . У головній діагоналі матриці знайти перший від'ємний і останній додатний елементи, а також поміняти їх місцями.

### **Варіант 9**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках знайти в ній останній нульовий елемент і його місцезнаходження (координати).

### **Варіант 10**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному стовпчику матриці знайти останній максимальний елемент і його місцезнаходження (координати).

### **Варіант 11**

Задано матрицю дійсних чисел  $A[n,n]$ . У побочній діагоналі матриці знайти перший максимальний і останній мінімальний елементи, а також поміняти їх місцями.

### **Варіант 12**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по стовпчиках знайти в ній останній мінімальний елемент і його місцезнаходження (координати).

### **Варіант 13**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному рядку матриці знайти перший максимальний елемент і його місцезнаходження (координати).

### **Варіант 14**

Задано матрицю дійсних чисел  $A[n,n]$ . У головній діагоналі матриці знайти перший мінімальний і останній максимальний елементи, а також поміняти їх місцями.

### **Варіант 15**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках визначити в ній присутність заданого дійсного числа  $X$  і його місцезнаходження (координати).

### **Варіант 16**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному стовпчику матриці знайти перший від'ємний елемент і його місцезнаходження (координати).

### **Варіант 17**

Задано матрицю дійсних чисел  $A[n,n]$ . У побочній діагоналі матриці знайти перший додатний і останній від'ємний елементи, а також поміняти їх місцями.

### **Варіант 18**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по стовпчиках визначити в ній присутність заданого дійсного числа  $X$  і його місцезнаходження (координати).

### **Варіант 19**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному рядку матриці знайти перший нульовий елемент і його місцезнаходження (координати).

### **Варіант 20**

Задано матрицю дійсних чисел  $A[n,n]$ . У головній діагоналі матриці знайти перший максимальний і останній мінімальний елементи, а також поміняти їх місцями.

### **Варіант 21**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках знайти в ній останній максимальний елемент і його місцезнаходження (координати).

### **Варіант 22**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному стовпчику матриці знайти останній додатний елемент і його місцезнаходження (координати).

### **Варіант 23**

Задано матрицю дійсних чисел  $A[n,n]$ . У побочній діагоналі матриці знайти перший від'ємний і останній додатний елементи, а також поміняти їх місцями.

### **Варіант 24**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по стовпчиках знайти в ній перший від'ємний елемент і його місце - знаходження (координати).

### **Варіант 25**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному рядку матриці знайти останній мінімальний елемент і його місцезнаходження (координати).

### **Варіант 26**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по стовпчиках знайти в ній останній додатний елемент і його місцезнаходження (координати).

### **Варіант 27**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках знайти в ній перший максимальний елемент і його місцезнаходження (координати).

### **Варіант 28**

Задано матрицю дійсних чисел  $A[m,n]$ . У кожному стовпчику матриці знайти перший мінімальний елемент і його місцезнаходження (координати).

### **Варіант 29**

Задано матрицю дійсних чисел  $A[n,n]$ . У побочній діагоналі матриці визначити присутність заданого дійсного числа  $X$  і його місцезнаходження (координати).

### **Варіант 30**

Задано матрицю дійсних чисел  $A[m,n]$ . При обході матриці по рядках знайти в ній останній мінімальний елемент і його місцезнаходження (координати).



## **6. ЛАБОРАТОРНА РОБОТА №1.6.**

### **АЛГОРИТМИ ОБХОДУ ДВОВИМІРНИХ МАСИВІВ (МАТРИЦЬ)**

#### **Мета лабораторної роботи**

Метою лабораторної роботи №1.6. «Алгоритми обходу двовимірних масивів (матриць)» є засвоєння теоретичного матеріалу та набуття практичних навичок гнучкої роботи при реалізації алгоритмів з використанням двовимірних масивів (матриць).

#### **Постановка задачі**

1. Оскільки екран монітора має площинні координати так само, як і двовимірний масив (матриця), але, на відміну від останнього, надає можливість візуально спостерігати виконання способу обходу, дана лабораторна робота виконується в координатах екрану монітора (дивись методичні вказівки до виконання даної лабораторної роботи).

2. Завданням даної лабораторної роботи є виконання заданого за варіантом способу обходу на екрані монітору в текстовому режимі, проставляючи довільний символ клавіатури (наприклад ‘\*’) у порядку заданого способу обходу.

3. Оскільки при виводі символу у правий ніжній кут екрану відбувається зсув зображення на один рядок вгору (якщо тільки не використовується прямиий доступ до відеопам’яті), *останній рядок екрану монітора при виконанні завдання заповнювати не треба.*

## Зміст звіту

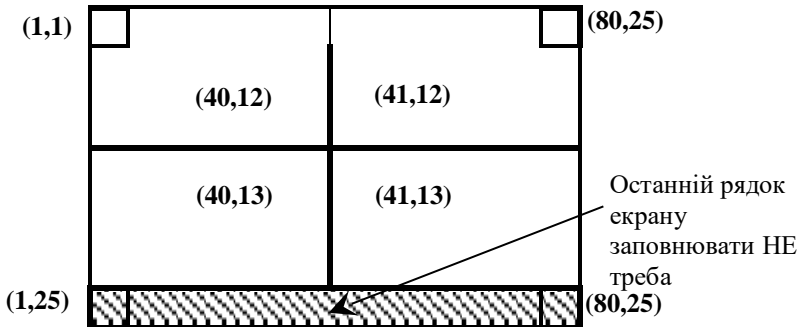
1. Загальна постановка задачі та завдання для конкретного варіанту з рисунком заданого обходу двовимірного масиву (матриці).
2. Текст програми.
3. Тестування програми. При тестуванні необхідно уважно відстежити правильність заповнення екрану монітора символами (виконання обходу матриці), при повільному виводі символів (використовуючи функції затримки виконання програми).
4. Окрім текстового звіту, в якості результату, продемонструвати викладачу правильність роботи програми при повільному виводі символів.

## Контрольні питання

1. Класифікація масивів.
2. Запис оголошення та використання двовимірних масивів діаграмами дій.
3. Запис оголошення та використання двовимірних масивів мовою програмування.
4. Що означає термін «обхід двовимірного масиву (матриці)»?
5. Скільки циклів та індексних змінних потрібно для виконання обходу двовимірного масиву (матриці)?
6. Як визначити спосіб простого обходу (по рядках чи по стовпчиках), використаного у програмі, за виглядом заголовків циклів та використання індексних змінних при звертанні до елементів масиву?

## Методичні вказівки до виконання завдання

1. Екран монітора комп'ютера у стандартному текстовому режимі має 25 рядків і 80 стовпчиків. Координати кутових та центральних (якщо ширину екрану вважати рівною 24 рядкам) позицій екрану монітора комп'ютера показані на наступному рисунку,



При виконанні завдання даної лабораторної роботи, при правильно складених алгоритмах, зображення на екрані монітора буде автоматично «підніматись» на один рядок вгору в момент виводу символу у правий нижній кут екрану з координатами (80,25).

*Тому, при складанні алгоритму останній рядок екрану монітора при виконанні завдання заповнювати не треба, тобто останнім рядком вважати 24-й рядок.*

Потрібно зауважити, що можна написати правильну програму, яка буде коректно заповнювати і 25-й рядок, але для цього треба використовувати прямий доступ до відеопам'яті комп'ютера).

2. Треба звернути увагу на протилежний спосіб нумерації (відкладення) координат у двовимірному масиві (матриці) та на екрані монітора.

При звертанні до елемента двовимірного масиву  $A[i, j]$ , перший індекс  $i$  є номером рядка і відкладається по вертикалі (вниз), а другий індекс  $j$  є номером стовпчика і відкладається по горизонталі (вправо).

При звертанні до позиції екрану монітора (найчастіше за допомогою процедури/функції типу *gotoxy(X,Y)*, **навпаки**, перший індекс  $X$  є номером стовпчика і відкладається по горизонталі (вправо), а другий індекс  $Y$  є номером рядка і відкладається по вертикалі (вниз).

З врахуванням цієї відмінності, в усіх інших аспектах алгоритми реалізації обходу двовимірного масиву (матриці) та обходу позицій екрану монітора будуть однаковими.

3. Враховуючи високу швидкодію сучасних комп'ютерів екран монітора буде заповнюватись миттєво, в результаті чого відстежити коректність виконання обходу неможливо без використання в програмі процедур або функцій затримки виконання програми на певний проміжок часу після виводу кожного наступного символу.

У мовах програмування процедура/функція/метод затримки, як правило, має один параметр, який визначає тривалість затримки виконання програми у мілісекундах.

У мовах Turbo Pascal та Free Pascal така процедура має ідентифікатор *Delay* і знаходиться у модулі *Crt*.

У мові C для затримки використовується функція *usleep()*, яка вимагає підключення заготовочного файлу *<unistd.h>*.

Для виконання затримки у мові C++ можуть використовуватись функції/методу *Sleep()*, *sleep()* або *delay()* (Borland C++ Builder 3.1), підключення яких є різним, в залежності від компілятора мови C++ та операційної системи (Windows, Linux, MSDOS).

Затримка у мові C# виконується за допомогою методу *Thread.Sleep()*, для використання якого попередньо треба підключити модуль *System.Threading*.

У мові Java затримка виконується так само, як у мові C#, з тією відмінністю, що для її використання додаткові оператори підключення не потрібні.

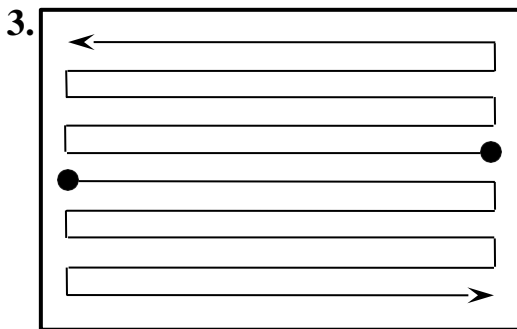
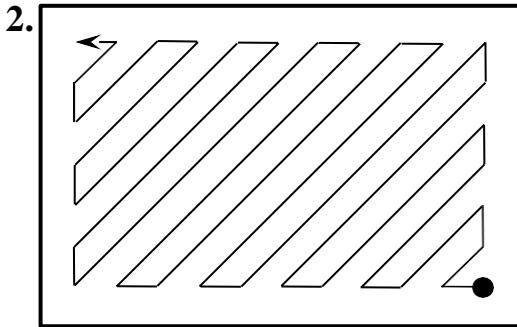
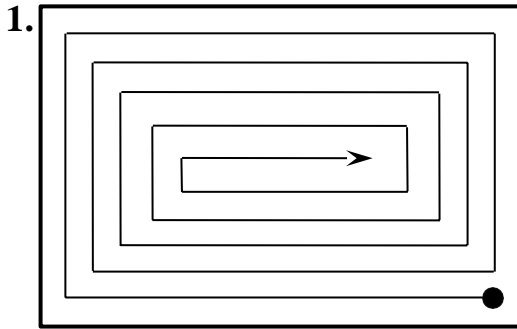
### **Умовні позначення на схемах варіантів завдань**

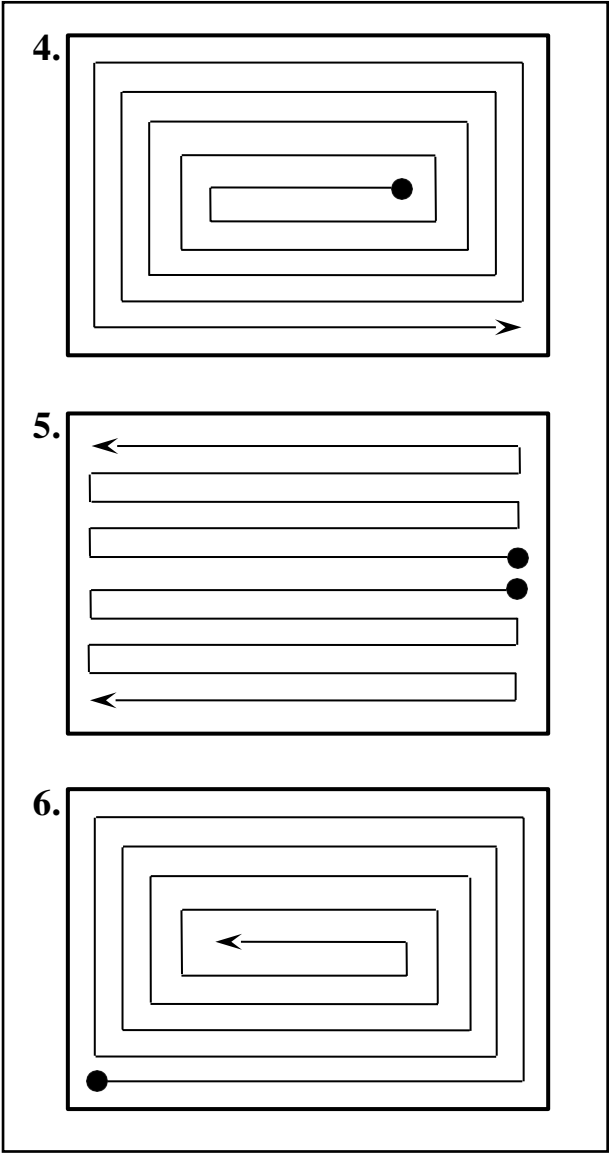
1. Якщо на схемі завдання зображена тільки одна спіралеподібна чи зигзагоподібна стрілка, то символи на екран повинні бути виведені строго згідно зміні напрямків лінії стрілки та її перегибів.

2. Якщо на схемі завдання зображені дві зигзагоподібні стрілки, то символи на екран повинні бути виведені строго згідно

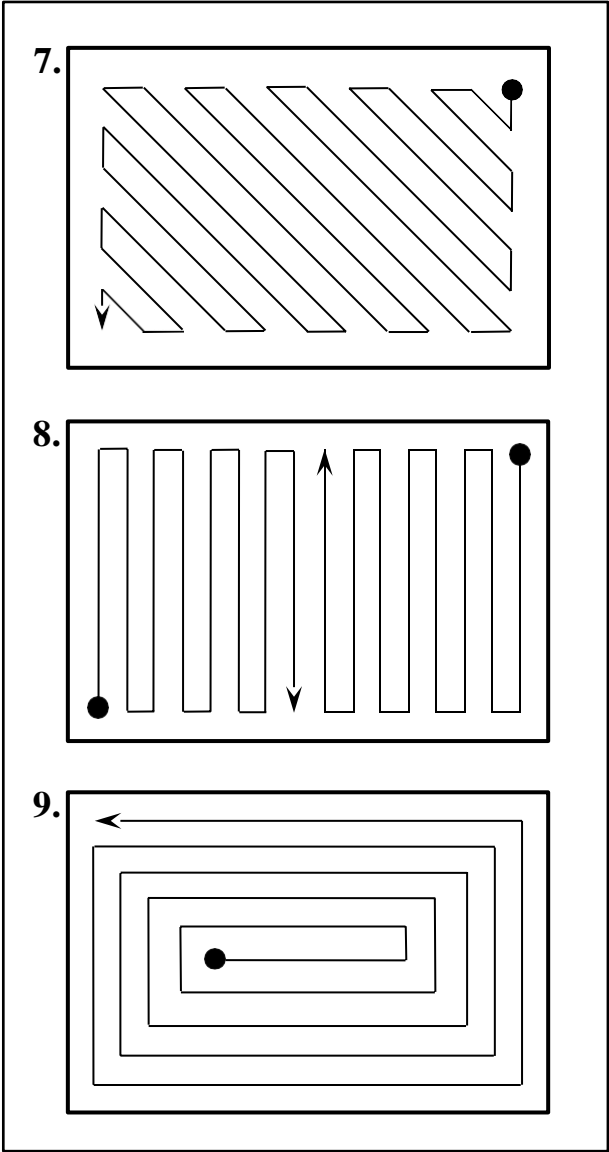
зміні напрямків лінії стрілки та її перегинів *почергово*, тобто за принципом «перший символ згідно першої стрілки, перший символ згідно другої стрілки, другий символ згідно першої стрілки, другий символ згідно другої стрілки, і т.д.».

### **Варіанти індивідуальних завдань**

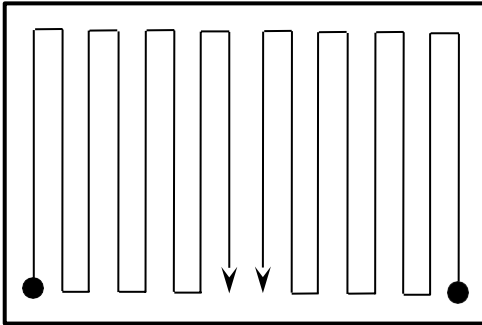




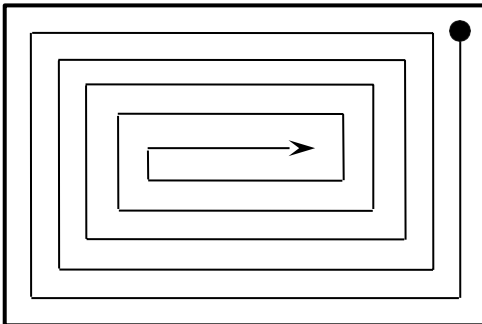




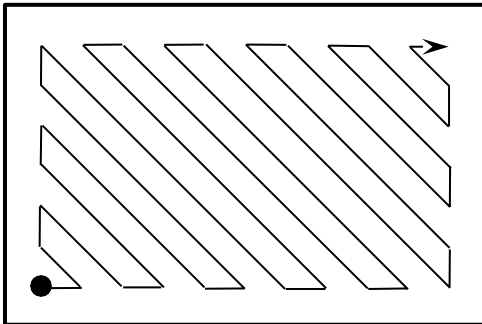
10.



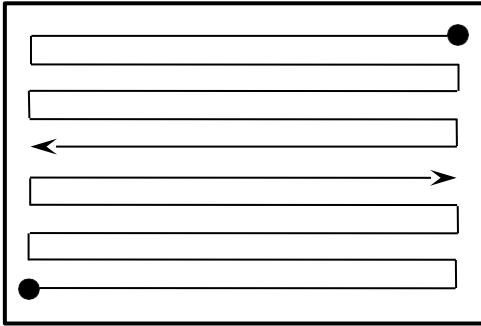
11.



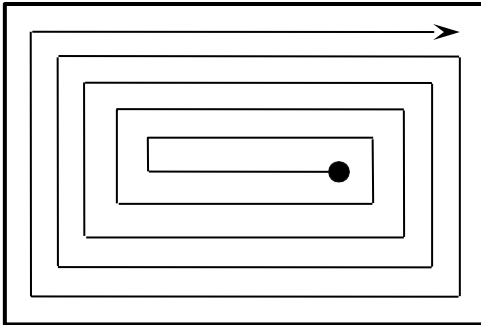
12.



13.



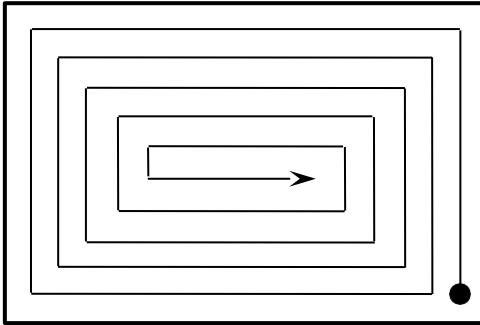
14.



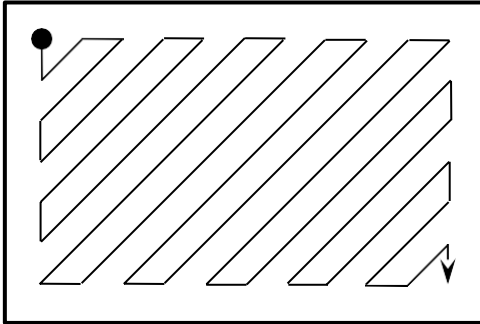
15.



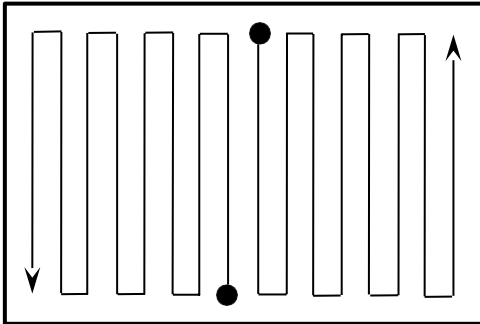
16.



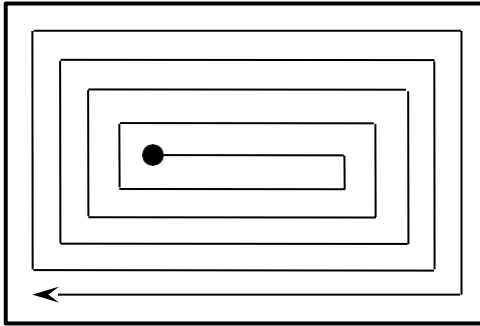
17.



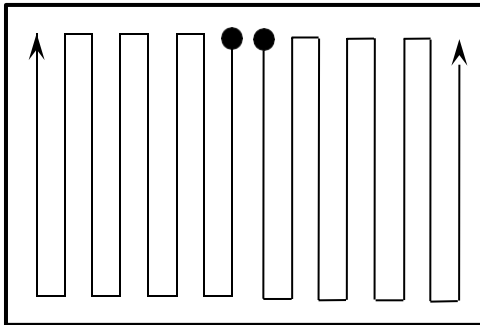
18.



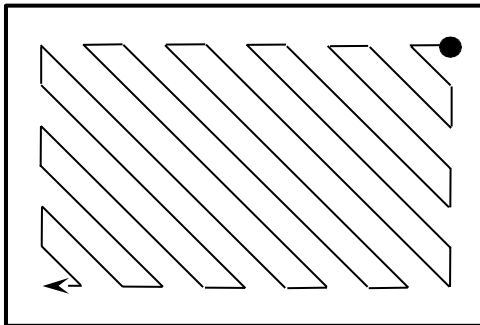
19.



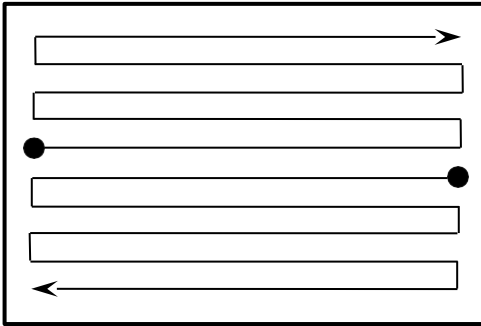
20.



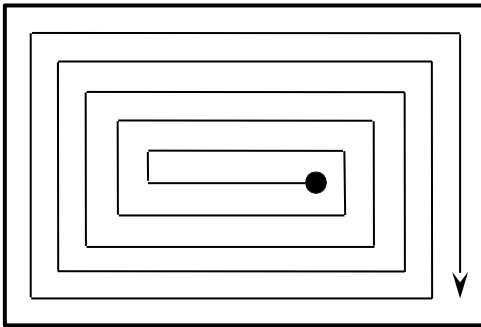
21.



22.



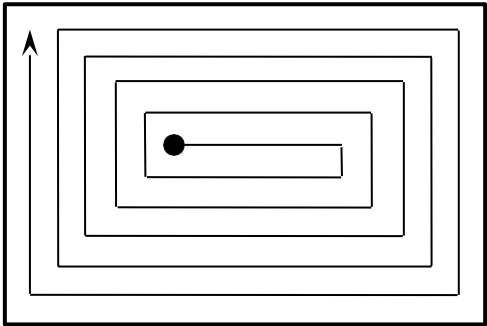
23.



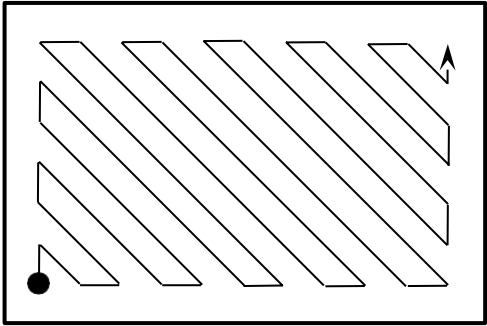
24.



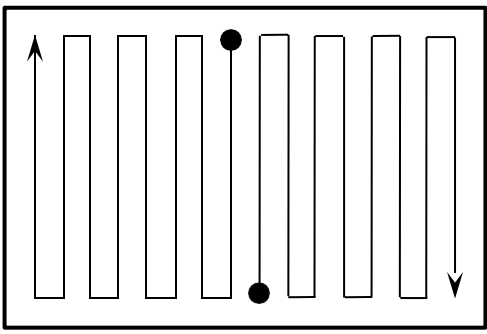
25.



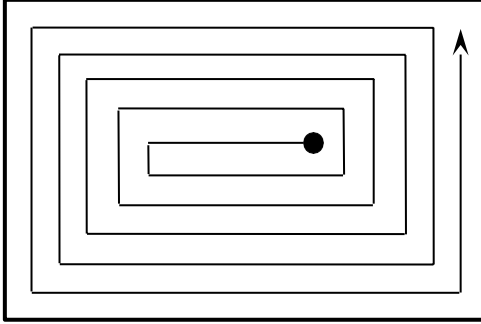
26.



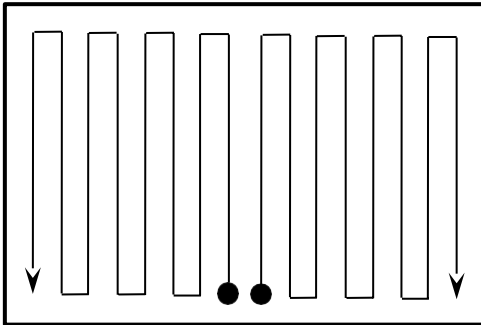
27.



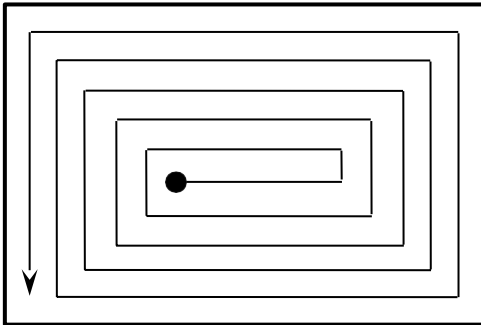
28.



29.



30.





## СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

### Основна література

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989.
2. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0. – 9-е изд. – К.: Век+, Спб.: КОРОНА-Век, 2007. – 464 с., ил.
3. Т.Кормен, Ч.Лейзерзон, Р.Ривест. Алгоритмы: построение и анализ. М.: МЦНМО, 2000.
4. Кнут Д. Искусство программирования для ЭВМ. т.1. Основные алгоритмы., М.: Мир, 1976.
5. Кнут Д. Искусство программирования для ЭВМ. т.2. Полученные алгоритмы., М.: Мир, 1977.
6. Кнут Д. Искусство программирования для ЭВМ. т.3. Сортировка и поиск., М.: Мир, 1978.

### Додаткова література

7. Вирт Н. Алгоритмы + структуры данных = программы. М.:Мир, 1985.
8. Г.Буч. Объектно-ориентированное проектирование с примерами применения: Пер. с англ. – М.: Конкорд, 1992. – 519 с., ил.
9. Методичні вказівки для виконання лабораторних робіт з дисципліни „Програмування” для студентів спеціальностей „Комп’ютерні системи та мережі”, „Спеціалізовані комп’ютерні системи”, „Системне програмування” , ч. 1,2 (Укл. Р.Ф. Колінько, О.І. Марченко). Київ НТУУ „КПІ” – 2004, видавництво „Ювета”.

10.Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн, М.И. Задачи по программированию., М.: наука, 1988.

11. Martin J., McClure C. Structured techniques : The basis for CASE., 1988.

12. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов., М.: Мир, 1979.

13. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов., М.: Мир, 1981.

14. Дал У., Дейкстра., Хоор К. Структурное программирование., М.: Мир, 1975.

15. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978.

16. Вирт Н. Систематическое программирование. Введение., М.: Мир,1977.

17. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования., М.: Мир, 1982.

18. Брукс Ф.П. Как проектируются и создаются программные комплексы., М.: Наука, 1979.