

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ

Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

# Основи комп'ютерних технологій

## Методичні рекомендації

до виконання лабораторних робіт

*для здобувачів денної та заочної форм навчання спеціальностей*

*122 "Комп'ютерні науки ", 123 "Комп'ютерна інженерія" та*

*125 "Кібербезпека"*

ЗАТВЕРДЖЕНО

на засіданні кафедри кібербезпеки та програмного  
забезпечення,

протокол від 15 лютого 2022 року № 12

Кропивницький 2022

**Основи комп'ютерних технологій:** метод. рекомендації до викон. лаб. робіт для здобувач. денної та заочної форм навч. спец. 122 "Комп'ютерні науки", 123 "Комп'ютерна інженерія", 125 "Кібербезпека"/ уклад. О.С. Улічев, Л. І. Поліщук, К.О. Буравченко. — Кропивницький: ЦНТУ, 2022. — 57 с.

Укладачі: Улічев О.С.,  
Поліщук Л. І.,  
Буравченко К.О.

Рецензенти: Смірнов О. А., д-р техн. наук, професор;  
Доренський О. П., канд. техн. наук, доцент

© Улічев О.С., Поліщук Л. І., Буравченко К.О.,  
укладання, 2022

© Центральноукраїнський національний  
технічний університет, 2022

## Зміст

	Стор.
Вступ.....	4
Лабораторна робота №1. Системи числення.....	5
Лабораторна робота №2. Базові поняття. Алгоритмічні конструкції.....	11
Лабораторна робота №3. Обробка масивів та рядків.....	28
Лабораторна робота №4. Алгоритми сортування та пошуку.....	45
Лабораторна робота №5. Основи об'єктного програмування.....	52
Додатки.....	55

## Вступ

Курс «Основи комп'ютерних технологій» є обов'язковий для здобувачів кафедри “Кібербезпеки та програмного забезпечення”. Фактично він є вступним оглядовим курсом з основ програмування і має на меті введення до спеціальності комп'ютерних технологій.

Дані методичні рекомендації адресовані майбутнім розробникам програмного забезпечення та фахівцям з аналізу даних, також можуть використовуватися здобувачами інших спеціальностей. Вони включають в себе основні теоретичні положення та практичні завдання, необхідні для засвоєння навчального матеріалу. А також містять основні поняття алгоритмізації та програмування для напрацювання та закріплення базових практичних навичок розробки програм з використанням основної мови програмування C#.

Здобувачам необхідно ознайомитися з теоретичним матеріалом, відповісти на питання та виконати всі завдання, що даються у лабораторних роботах.

Виконання і оформлення звітів здійснюється у відповідності до вимог, встановлених державними стандартами, перерахованими в таблиці 1.

Таблиця 1 – Державні стандарти, що слід застосовувати при оформленні звітів

Найменування документа	Стандарт
1 Пояснювальна записка	ДСТУ 1.5:2003 Правила побудови, викладення, оформлення та вимоги до змісту нормативних документів ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення ДСТ 2.104-2006 ЄСКД. Основні написи ДСТ 2.105-95 ЄСКД. Загальні вимоги до текстових документів ДСТ 2.106-96 ЄСКД. Текстові документи ДСТ 23501.4-79 САПР. Загальні вимоги до програмного забезпечення ДСТ 24.203-80 АСУ. Вимоги до змісту загальносистемних документів ДСТ 24.301-80 АСУ. Загальні вимоги до виконання текстових документів ДСТ 19.106-78 ЄСПД. Вимоги до програмних документів, виконаних друкованим чином ДСТ 19.404-79 ЄСПД. Пояснювальна записка. Вимоги до змісту і оформлення
2 Специфікація	ДСТ 2.106-96 ЄСКД. Текстові документи ДСТ 19.202-78 ЄСПД. Специфікація. Вимоги до змісту і оформлення
3 Креслення	ДСТ 2.417-78 ЄСКД. Правила виконання креслень друкованих плат ДСТ 24.304-82 АСУ. Вимоги до виконання креслень
4 Схеми	ДСТ 2.701-76 ЄСКД. Схеми. Вигляд і типи. Загальні вимоги до виконання ДСТ 19.701-90 ЄСПД. Схеми алгоритмів і програм. Позначки умовні графічні
5 Програми і програмні документи	ДСТ 19.101-77 ЄСПД. Вигляд програм і програмних документів ДСТ 19.105-78 ЄСПД. Загальні вимоги до програмних документів ДСТ 19.401-78 ЄСПД. Текст програми ДСТ 19.402-78. Опис програми
6 Опис	ДСТ 24.204-80 АСУ. Опис постановки задачі
7 Звіт	ДСТ 7.32-2001 ССІБВС. Звіт про науково-дослідницьку роботу. Структура і правила оформлення

## Лабораторна робота №1

### Тема: Системи числення

**Мета:** Розглянути системи числення, з якими працюють комп'ютери.  
Набути навичок при переході з однієї системи в іншу

### Теоретичні відомості

#### Алгоритми переходу між системами числення

В позиційній системі числення числа записуються таким чином, що кожен наступний (рухаючись з права на ліво) розряд більше попереднього на степінь основи системи числення.

#### Правила переведення чисел з однієї системи числення в іншу

Переведення чисел з однієї системи числення в іншу становить важливу частину машинної арифметики. Розглянемо основні правила переведення:

1. Для переведення двійкового числа в десяткове необхідно його записати у вигляді многочлена, що складається з цифр цього числа і відповідного степеню числа 2, і обчислити за правилами десятикової арифметики.

**Приклад.** Число перевести в десяткову систему числення.

$$10111,11_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 23,75_{10}$$

Такий многочлен використовують для перетворення числа із будь-якої системи в десяткове число, де основою є саме система, в якій це число наведено.

2. Для переведення десятикового числа в двійкову систему його необхідно послідовно ділити на 2 до тих пір, поки не залишиться залишок, менший або рівний 1. Число в двійковій системі записується як послідовність останнього результату ділення і залишків від ділення в зворотному порядку.

Для переведення дробової частини її необхідно множити на 2 стільки разів, скільки знаків після коми потрібно мати. Ціла частина від добутку і формує результат.

**Приклад.** Число перевести в двійкову систему числення з точністю до 3-х знаків після коми.

Handwritten division of 22 by 2 to get binary 10110:

$$\begin{array}{r|l} 22 & 2 \\ -22 & 11 \\ \hline 0 & 10 \\ & 5 \\ & 2 \\ & -4 \\ & 1 \\ & -2 \\ & 0 \end{array}$$

The remainders 1, 0, 1, 1, 0 are written below the division steps. A red arrow points from the final remainder 0 to the final binary result.

$$22,865_{10} = 10110,110_2$$

Handwritten multiplication of decimal parts by 2 to get binary 1,46,0,92:

$\begin{array}{r} 0,865 \\ * 2 \\ \hline 1,730 \end{array}$	$\begin{array}{r} 0,73 \\ * 2 \\ \hline 1,46 \end{array}$	$\begin{array}{r} 0,46 \\ * 2 \\ \hline 0,92 \end{array}$
---	---	---

A red arrow points from the final result 0,92 to the final binary result.

3. Для переведення десяткового числа в вісімкову систему його необхідно послідовно ділити на 8 до тих пір, поки не залишиться залишок, менший або рівний 7. Число в вісімковій системі записується як послідовність цифр останнього результату ділення і залишків від ділення в зворотному порядку.

Для переведення дробової частини її необхідно множити на 8 стільки разів, скільки знаків після коми потрібно мати. Ціла частина від добутку і формує результат.

**Приклад.** Число перевести в вісімкову систему числення з точністю до 3-х знаків після коми.

$  \begin{array}{r l}  571 & 8 \\  \hline  -56 & 71 \\  \hline  11 & -64 \\  \hline  -8 & 7 \\  \hline  3 &   \end{array}  $	$  \begin{array}{r l}  8 & 8 \\  \hline  -8 & 0 \\  \hline  0 &   \end{array}  $	$  \begin{array}{r l}  8 & 8 \\  \hline  -8 & 1 \\  \hline  1 &   \end{array}  $
--	--	--

$$\begin{array}{r}
 0,524 \\
 * \quad 8 \\
 \hline
 4,192
 \end{array}$$

$$\begin{array}{r}
 0,192 \\
 * \quad 8 \\
 \hline
 1,536
 \end{array}$$

$$\begin{array}{r}
 0,536 \\
 * \quad 8 \\
 \hline
 4,288
 \end{array}$$

$$571,524_{10} = 1073,414_8$$

4. Для переведення десяткового числа в шістнадцяткову систему його необхідно послідовно ділити на 16 до тих пір, поки не залишиться залишок, менший або рівний 15. Число в шістнадцятковій системі записується як послідовність цифр останнього результату ділення і залишків від ділення в зворотному порядку. Слід враховувати, що залишки більші за 9 позначають буквами латинського алфавіту (10-A, 11-B, 12-C, 13-D, 14-E, 15-F).

Для переведення дробової частини її необхідно множити на 16 стільки разів, скільки знаків після коми потрібно мати. Ціла частина від добутку і формує результат.

**Приклад.** Число перевести в шістнадцяткову систему числення з точністю до 3-х знаків після коми.

$  \begin{array}{r l}  7467 & 16 \\  \hline  -7458 & 466 \\  \hline  11 & -464 \\  \hline  2 & 29 \\  \hline  13 & -16 \\  \hline  1 &   \end{array}  $	$  \begin{array}{r}  0,819 \\  * \quad 16 \\  \hline  13,104  \end{array}  $	$  \begin{array}{r}  0,104 \\  * \quad 16 \\  \hline  1,664  \end{array}  $	$  \begin{array}{r}  0,664 \\  * \quad 16 \\  \hline  10,624  \end{array}  $
---	--	---	--

$$7467,219_{10} = 1D2B,D1A_{16}$$

Даний механізм використовують для переведення десяткового числа в будь-яку систему.

5. Щоб перевести число з двійкової системи в вісімкову, його потрібно розбити на тріади (трійки цифр), починаючи від десяткової коми, в

разі необхідності доповнити крайні тріади нулями, і кожену тріаду замінити відповідною вісімковою цифрою за таблицею 2.

Таблиця 2 – Метод тріад

8-ва система	0	1	2	3	4	5	6	7
2-ва система	000	001	010	011	100	101	110	111

**Приклад.** Число  $1011100011,01101_2$  перевести в вісімкову систему числення.

$$001\ 011\ 100\ 011,011\ 010_2 = 1343,32_8$$

6. Щоб перевести число з двійкової системи в шістнадцяткову, його потрібно розбити на тетради (четвірки цифр), починаючи від десяткової коми, в разі необхідності доповнити тетради нулями, і кожену тетраду замінити відповідною шістнадцятковою цифрою за таблицею 3.

Таблиця 3 – Метод тетрад

16-ва система	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2-ва система	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

**Приклад.** Число  $10111010011,0111011_2$  перевести в шістнадцяткову систему числення.

$$0101\ 1101\ 0011,0111\ 0110_2 = 5D3,76_{16}$$

7. Щоб перевести число з вісімкової системи в двійкову, достатньо записати кожену цифру вісімкового числа в вигляді двійкової тріади відповідно таблиці 2.

**Приклад.** Число  $432,15_8$  перевести в двійкову систему числення.

$$432,15_8 = 100\ 011\ 010\ ,\ 001\ 101_2$$

8. Щоб перевести число з шістнадцяткової системи в двійкову, достатньо записати кожену цифру шістнадцяткового числа в вигляді двійкової тетради відповідно таблиці 3.

**Приклад.** Число  $58C,B2_{16}$  перевести в двійкову систему числення.

$$58C,B2_{16} = 0101\ 1000\ 1100\ ,\ 1011\ 0010_2$$

**Завдання:** здійснити перетворення з однієї системи числення в іншу у відповідності з варіантом з точністю до 4-го знаку після коми. Звіт до лабораторної роботи повинен містити: титульний аркуш (назва та номер ЛР, дані автора), мету роботи, індивідуальні завдання (завдання за варіантом), відповіді на контрольні запитання, висновки (див. Додатки А, Б).

**Варіант №1**

1.  $57,36_8 \rightarrow ?(2)$
2.  $11010,101_2 \rightarrow ?(10)$
3.  $32,16_8 \rightarrow ?(10)$
4.  $8B,12_{16} \rightarrow ?(10)$
5.  $11100011111,1101101_2 \rightarrow ?(16)$
6.  $6F1A,84_{16} \rightarrow ?(2)$
7.  $396,64_{10} \rightarrow ?(16)$
8.  $538,64_{10} \rightarrow ?(2)$
9.  $286,52_{10} \rightarrow ?(8)$
10.  $1000100111100,100101_2 \rightarrow ?(8)$

**Варіант №2**

1.  $67,54_8 \rightarrow ?(2)$
2.  $11011,11_2 \rightarrow ?(10)$
3.  $864,85_{10} \rightarrow ?(16)$
4.  $111011011,111011_2 \rightarrow ?(16)$
5.  $1A2C,23_{16} \rightarrow ?(2)$
6.  $10111011001,0101111_2 \rightarrow ?(8)$
7.  $891,63_{10} \rightarrow ?(2)$
8.  $169,74_{10} \rightarrow ?(8)$
9.  $4A,C1_{16} \rightarrow ?(10)$
10.  $362,35_8 \rightarrow ?(10)$

**Варіант №3**

1.  $37,64_8 \rightarrow ?(2)$
2.  $10101,101_2 \rightarrow ?(10)$
3.  $8B,2D_{16} \rightarrow ?(10)$
4.  $1110111011111,1101101_2 \rightarrow ?(16)$
5.  $2C4A,56_{16} \rightarrow ?(2)$
6.  $321,82_{10} \rightarrow ?(16)$
7.  $284,66_{10} \rightarrow ?(8)$
8.  $1011101110,111101111011_2 \rightarrow ?(8)$
9.  $649,94_{10} \rightarrow ?(2)$
10.  $356,64_8 \rightarrow ?(10)$

**Варіант №4**

1.  $162,72_8 \rightarrow ?(2)$
2.  $101101,1001_2 \rightarrow ?(10)$
3.  $7A1,4C_{16} \rightarrow ?(10)$
4.  $1111011100,1010101_2 \rightarrow ?(16)$
5.  $2B4C,82_{16} \rightarrow ?(2)$
6.  $454,64_{10} \rightarrow ?(16)$
7.  $735,24_8 \rightarrow ?(10)$
8.  $296,82_{10} \rightarrow ?(2)$
9.  $327,85_{10} \rightarrow ?(8)$
10.  $11101110,110111_2 \rightarrow ?(8)$

**Варіант №5**

1.  $561,24_{10} \rightarrow ?(8)$
2.  $11,6_{16} \rightarrow ?(10)$
3.  $1110001110,11010001111_2 \rightarrow ?(16)$
4.  $1060,54_8 \rightarrow ?(2)$
5.  $56,1875_{10} \rightarrow ?(2)$
6.  $11111001,01001_2 \rightarrow ?(10)$
7.  $2A6,9B_{16} \rightarrow ?(2)$
8.  $2015,4317_8 \rightarrow ?(10)$
9.  $10111,0101_2 \rightarrow ?(8)$
10.  $19,75_{10} \rightarrow ?(16)$

**Варіант №6**

1.  $762,534_{10} \rightarrow ?(8)$
2.  $61C,8B_{16} \rightarrow ?(10)$
3.  $425,61_8 \rightarrow ?(2)$
4.  $10111,011_2 \rightarrow ?(10)$
5.  $64,82_{10} \rightarrow ?(2)$
6.  $874,62_{10} \rightarrow ?(16)$
7.  $614,54_8 \rightarrow ?(10)$
8.  $2E1,4A_{16} \rightarrow ?(2)$
9.  $11100,001_2 \rightarrow ?(8)$
10.  $10001,1111_2 \rightarrow ?(16)$

**Варіант №7**

1.  $213,74_{10} \rightarrow ?(8)$
2.  $2A1,23_{16} \rightarrow ?(10)$
3.  $311,51_8 \rightarrow ?(2)$
4.  $11101,101_2 \rightarrow ?(10)$
5.  $92,25_{10} \rightarrow ?(2)$
6.  $759,9375_{10} \rightarrow ?(16)$
7.  $421,7_8 \rightarrow ?(10)$
8.  $1C4,27_{16} \rightarrow ?(2)$
9.  $11011,11_2 \rightarrow ?(8)$
10.  $1110,01_2 \rightarrow ?(16)$



**Варіант №8**

1.  $251,64_{10} \rightarrow ?(8)$
2.  $4C8,1D_{16} \rightarrow ?(10)$
3.  $564,71_8 \rightarrow ?(2)$
4.  $10010,1011_2 \rightarrow ?(10)$
5.  $44,95_{10} \rightarrow ?(2)$

6.  $742,34_{10} \rightarrow ?(16)$
7.  $624,15_8 \rightarrow ?(10)$
8.  $1A2,A4_6 \rightarrow ?(2)$
9.  $10001,1011_2 \rightarrow ?(8)$
10.  $10011,1111_2 \rightarrow ?(16)$

**Варіант №9**

1.  $518,64_{10} \rightarrow ?(8)$
2.  $1F1,6E_{16} \rightarrow ?(10)$
3.  $413,32_8 \rightarrow ?(2)$
4.  $11010,011_2 \rightarrow ?(10)$
5.  $83,25_{10} \rightarrow ?(2)$

6.  $431,58_{10} \rightarrow ?(16)$
7.  $512,43_8 \rightarrow ?(10)$
8.  $2A3,F1_{16} \rightarrow ?(2)$
9.  $10010,01_2 \rightarrow ?(8)$
10.  $10111,101_2 \rightarrow ?(16)$

**Варіант №10**

1.  $541,26_{10} \rightarrow ?(8)$
2.  $1C4,5A_{16} \rightarrow ?(10)$
3.  $476,51_8 \rightarrow ?(2)$
4.  $10101,101_2 \rightarrow ?(10)$
5.  $64,85_{10} \rightarrow ?(2)$

6.  $124,67_{10} \rightarrow ?(16)$
7.  $254,36_8 \rightarrow ?(10)$
8.  $5A,2D_{16} \rightarrow ?(2)$
9.  $10010,001_2 \rightarrow ?(8)$
10.  $10111,01_2 \rightarrow ?(16)$

**Варіант №11**

1.  $482,73_{10} \rightarrow ?(8)$
2.  $8B,64_{16} \rightarrow ?(10)$
3.  $712,53_8 \rightarrow ?(2)$
4.  $10011,1101_2 \rightarrow ?(10)$
5.  $35,27_{10} \rightarrow ?(2)$

6.  $648,53_{10} \rightarrow ?(16)$
7.  $327,35_8 \rightarrow ?(10)$
8.  $3C1,A8_{16} \rightarrow ?(2)$
9.  $100111011,1_2 \rightarrow ?(8)$
10.  $1000111110,11111_2 \rightarrow ?(16)$

**Варіант №12**

1.  $629,54_{10} \rightarrow ?(8)$
2.  $8E4,1B_{16} \rightarrow ?(10)$
3.  $457,16_8 \rightarrow ?(2)$
4.  $10101,011_2 \rightarrow ?(10)$
5.  $86,52_{10} \rightarrow ?(2)$

6.  $432,96_{10} \rightarrow ?(16)$
7.  $513,72_8 \rightarrow ?(10)$
8.  $2A5,1B_{16} \rightarrow ?(2)$
9.  $10011,11_2 \rightarrow ?(8)$
10.  $11101,101_2 \rightarrow ?(16)$

**Варіант №13**

1.  $56,81_{10} \rightarrow ?(8)$
2.  $18E,2B_{16} \rightarrow ?(10)$
3.  $541,27_8 \rightarrow ?(2)$
4.  $11000,001_2 \rightarrow ?(10)$
5.  $54,28_{10} \rightarrow ?(2)$

6.  $841,66_{10} \rightarrow ?(16)$
7.  $433,21_8 \rightarrow ?(10)$
8.  $2A8,6D_{16} \rightarrow ?(2)$
9.  $10001,111_2 \rightarrow ?(8)$
10.  $10010,001_2 \rightarrow ?(16)$

**Варіант №14**

1.  $562,41_{10} \rightarrow ?(8)$
2.  $68C,1B_{16} \rightarrow ?(10)$
3.  $432,07_8 \rightarrow ?(2)$
4.  $10111,011_2 \rightarrow ?(10)$
5.  $64,25_{10} \rightarrow ?(2)$

6.  $841,66_{10} \rightarrow ?(16)$
7.  $512,34_8 \rightarrow ?(10)$
8.  $2D1,4F_{16} \rightarrow ?(2)$
9.  $10110,011_2 \rightarrow ?(8)$
10.  $10011,11_2 \rightarrow ?(16)$

**Варіант №15**

- |                                    |                                      |
|------------------------------------|--------------------------------------|
| 1. $342,68_{10} \rightarrow ?(8)$  | 6. $896,68_{10} \rightarrow ?(16)$   |
| 2. $4D,8C_{16} \rightarrow ?(10)$  | 7. $612,43_8 \rightarrow ?(10)$      |
| 3. $471,62_8 \rightarrow ?(2)$     | 8. $2A8,F1_{16} \rightarrow ?(2)$    |
| 4. $10010,011_2 \rightarrow ?(10)$ | 9. $10010,101_2 \rightarrow ?(8)$    |
| 5. $54,68_{10} \rightarrow ?(2)$   | 10. $10111,1111_2 \rightarrow ?(16)$ |

**Контрольні запитання:**

1. Яка природна система числення комп'ютерів?
2. Яким чином записують многочлен для переведення в десяткову систему?
3. Який порядок перетворення десяткових чисел в будь-які інші?
4. В чому особливість використання шістнадцяткової системи числення?
5. Що таке тріада?
6. Що таке тетрада?
7. Який порядок перетворення шістнадцяткових і вісімкових чисел в двійкові?
8. Який порядок перетворення двійкових чисел в шістнадцяткові і вісімкові?

## Лабораторна робота №2

### Тема: Базові поняття. Алгоритмічні конструкції

**Мета:** Розглянути базові алгоритмічні конструкції для всіх мов програмування. Набути навичок в реалізації таких конструкцій операторами мови C#

#### Приклад 1. Розгалуження

##### Завдання

Проаналізувати умову задачі. Розробити алгоритм і скласти програму для обчислення. Дано  $x, y, z$  обчислити:

$$a = \ln \left| \left( y - \sqrt{x} \right) \left( x - \frac{y}{z + x^2/4} \right) \right|; \quad b = x - \frac{x^2}{3!} + \frac{x^5}{5!}.$$

##### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 2.1.

В завданні, при обчисленні  $A$  може виникати ряд моментів:

- $x$  - аргумент функції «корінь» має бути  $\geq 0$ ;
- $z + x * x / 4$  – знаменник, не може бути рівним 0.

Враховуючи таку необхідність додаємо до алгоритму перевірку умови (конструкцію розгалуження). Блок-схему наведено на рисунку 2.1 нижче.

Оскільки результатом обчислення виразу  $(z + x * x / 4)$  є число з плаваючою комою, то для надійного порівняння використовуємо порівняння з точністю 0.00000001 ( або «порівняння з епсилон»).

Тоді умова в алгоритмі  $(x < 0) \parallel ((z + x * x / 4) \neq 0)$  реалізується в програмі як:

$$((x < 0) \parallel ((z + x * x / 4) < 0.00000001))^1$$

##### Програмна реалізація

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            double a, b, x, y, z;
            Console.WriteLine("Завдання");
        }
    }
}
```

<sup>1</sup> Зверніть увагу на формування складної умови та спосіб порівняння дійсних чисел

```

        Console.WriteLine("Введіть параметри x (x>0), y,
z:");

        Console.Write("x=");
        x = Convert.ToDouble(Console.ReadLine());
        Console.Write("y=");
        y = Convert.ToDouble(Console.ReadLine());
        Console.Write("z=");
        z = Convert.ToDouble(Console.ReadLine());
        if ((x < 0) || ((z + x * x / 4) != 0))//перевірка
допустимих значень
        {
            Console.WriteLine("Обчислення неможливе при
даних параметрах");
        }
        else
        {
            double a1 = y - Math.Sqrt(x);
            double a2 = (x - y / (z + x * x / 4));
            a = Math.Log(Math.Abs(a1 * a2));
            b = x-Math.Pow(x,2)/6+Math.Pow(x,5)/120;
            Console.WriteLine("Результат:\n a={0}\n
b={1}", a,b);
        }
        Console.ReadLine();
    }
}

```

### Результати виконання програми

Працездатність програми перевірено із використанням контрольного прикладу.

Тест завдання де  $x=12$ ,  $y=1$ ,  $z=3$ . Результати роботи програми наведені на рисунку 2.2.

Правильність розрахунків виконаних в програмі, перевірено засобами Excel рисунок 2.3.

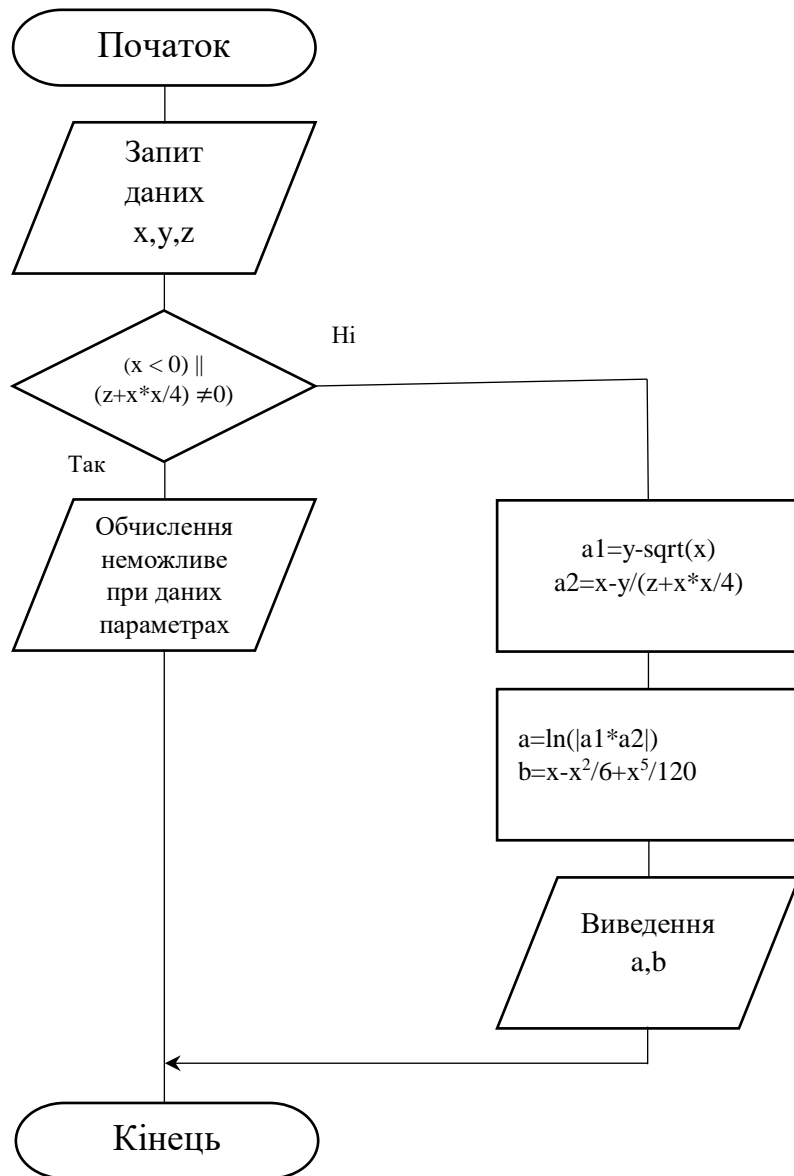


Рисунок 2.1 – Блок-схема алгоритму задачі

```

Завдання
Введіть параметри x (x>0), y, z:
x=12
y=1
z=3
Результат:
a=3,38459489626999
b=2061,6
  
```

Рисунок 2.2 – Результати виконання програми

x=	12
y=	1
z=	3
a=	3,384595
b=	2061,6

Рисунок 2.3 – Перевірка результату

## Приклад 2. Розгалуження із вкладеним умовним оператором

### Завдання

Відомо, що з чотирьох чисел a1, a2, a3, a4 тільки три є рівними між собою. Знайти номер числа, що не дорівнює іншим, та вивести значення.

### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 2.4.

### Програмна реалізація

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    class Program
    {
        static void Main(string[] args)
        {
            int a1, a2, a3, a4;
            Console.WriteLine("Завдання");
            Console.WriteLine("Відомо, що з чотирьох чисел a1, a2, a3, a4 тільки три є рівними\n" + "між собою. Знайти номер числа, що не дорівнює іншим, та вивести значення.");
            Console.WriteLine("Введіть параметри a1,a2, a3, a4:");

            //При введенні має бути три однакових числа і одне, що відрізняється
            Console.Write("a1=");
            a1 = Convert.ToInt32(Console.ReadLine());
            Console.Write("a2=");
            a2 = Convert.ToInt32(Console.ReadLine());
            Console.Write("a3=");
            a3 = Convert.ToInt32(Console.ReadLine());
            Console.Write("a4=");
            a4 = Convert.ToInt32(Console.ReadLine());
            if (a1==a2)//шукане число a3 або a4
            {
```

```

        if (a3 == a1) //шукане число a4
        {
            Console.Write("Шукане число a4={0}", a4);
        }
        else //шукане число a3
        {
            Console.Write("Шукане число a3={0}", a3);
        }
    }
    else//необхідно a1 порівняти з будь-яким a3 або a4
    {
        if (a1 == a3) //шукане число a2
        {
            Console.Write("Шукане число a2={0}", a2);
        }
        else
        {
            Console.Write("Шукане число a1={0}", a1);
        }
    }
    Console.ReadLine();
}
}
}

```

### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів.

Тест1: a1=12, a2=12, a3=12, a4=13. Результати роботи програми наведені на рисунку 2.5.

Тест2: a1=13, a2=12, a3=12, a4=12. Результати роботи програми наведені на рисунку 2.6.

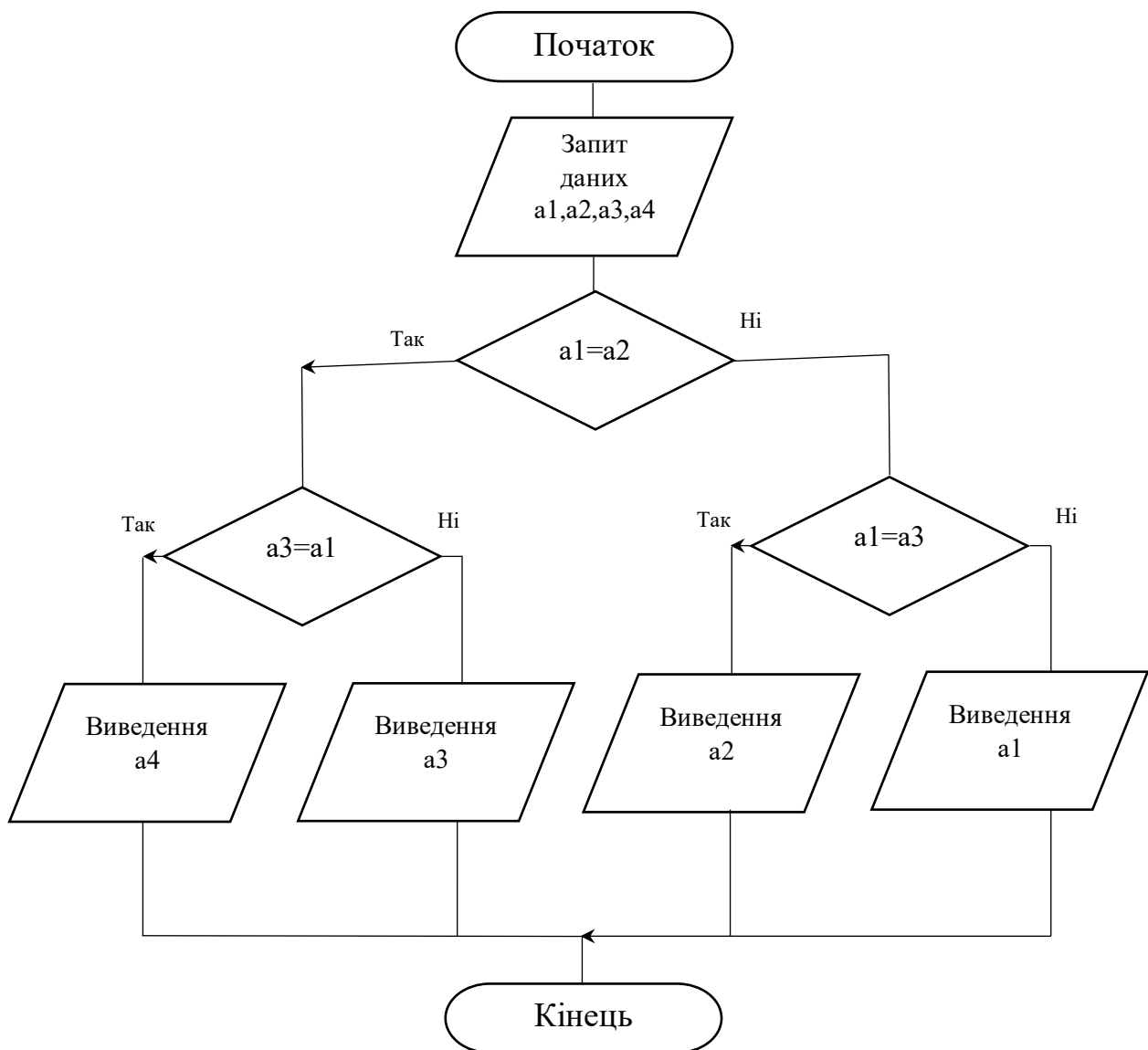


Рисунок 2.4 – Блок-схема алгоритму пошуку числа

```

Завдання
Відомо, що з чотирьох чисел a1, a2, a3, a4 тільки три є рівними
між собою. Знайти номер числа, що не дорівнює іншим, та вивести значення.
Введіть параметри a1,a2, a3, a4:
a1=12
a2=12
a3=12
a4=13
Шукане число a4=13
  
```

Рисунок 2.5 – Результати виконання програми Тест1



```

Завдання
Відомо, що з чотирьох чисел a1, a2, a3, a4 тільки три є рівними
між собою. Знайти номер числа, що не дорівнює іншим, та вивести значення.
Введіть параметри a1,a2, a3, a4:
a1=13
a2=12
a3=12
a4=12
Шукане число a1=13

```

Рисунок 2.6 – Результати виконання програми Тест 2

### Приклад 3. Цикл

#### Завдання

З заданою точністю  $\varepsilon$ , ( $0 < \varepsilon < 1$ ), знайти:

$$e^x = 1 + x/1! + x^2/2! + \dots + x^n/n! +$$

#### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 2.7.

#### Програмна реалізація<sup>2</sup>

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab3
{
    class Program
    {
        public static long factorial(int n)
        {
            if (n == 1)
            { return 1; }
            else
                return n * factorial(n - 1);
        }
        static void Main(string[] args)
        {
            double x, sum, si, e;
            int i = 1;
            Console.WriteLine("Завдання");
            Console.WriteLine("Обчислення експоненти за
            допомогою ряду з зад. точністю");
            Console.WriteLine("Введіть параметри x,e:");
            Console.Write("x=");
            x = Convert.ToDouble(Console.ReadLine());

```

<sup>2</sup> Зверніть увагу на форматування рядка виведення методу WriteLine і результати виведення в тестових запусках

```

        Console.WriteLine("e="); //точність
        e = Convert.ToDouble(Console.ReadLine());
        // Console.WriteLine("5!={0}", factorial(5));
//тест функції факторіал, в результаті буде закоментовано
        sum = 1; //початкове значення суми
        si = x; //i-й член
        Console.WriteLine(" i          i!          Si
Sum");
        Console.WriteLine(" 0          1          {0}
{1}", si, sum);
        while (si > e) //поки i-й член більше за необхідну
точність
        {
            sum = sum + si; //збільшуємо суму
            Console.WriteLine("{0}", String.Format("{0,2}{1,15}
{2,10:0.00000000}{3,10:0.00000000}", i, factorial(i), si, sum));
            i++;
            si = Math.Pow(x, i) / factorial(i); //знаходимо
новий доданок
        }
        Console.WriteLine("{0}", String.Format("{0,2}{1,10}
{2,10:0.00000000}{3,10:0.00000000}", i, factorial(i), si, sum));
        Console.WriteLine("Отримано з вик. ряду
EXP({0})={1}", x, sum);
        Console.WriteLine("Отримано з вик. станд. функції
EXP({0})={1}", x, Math.Exp(x));
        Console.ReadLine();
    }
}

```

## Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів.

Тест1:  $x=3$ ,  $e=0,001$  Результати роботи програми наведені на рисунку 2.8.

При обчисленні суми ряду використовується функція ФАКТОРІАЛ. Дана функція дуже швидко зростає, тому при великій кількості ітерацій (при малому значенні  $e$ ) будемо мати переповнення типу і невірний результат. Приклад такого випадку наведено на рисунку 2.9.

Тест2:  $x=3$ ,  $e=0,00001$  Результати роботи програми наведені на рисунку 2.9.

Тест 2 показує що при точності 0,00001 виникає переповнення і невірний результат. При зміні типу результату функції з `int` на `long` проблема вирішується (рисунок 2.10) та все рівно і даний тип може бути переповнено при великій кількості ітерацій. В лістингу наведено код вже з використанням типу `long`:

```
public static long factorial(int n)
```

Тест3:  $x=3$ ,  $e=0,000001$  Результати роботи програми наведені на рисунку 2.10.

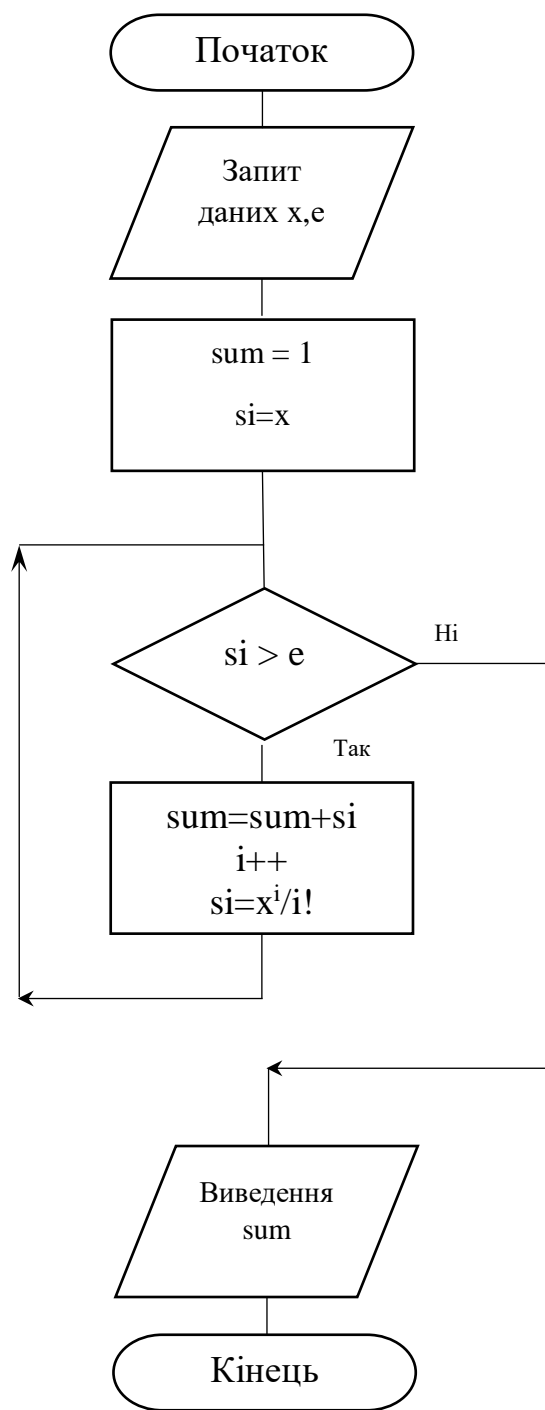


Рисунок 2.7 – Блок-схема обчислення суми ряду

```

Введіть параметри x,e:
x=3
e=0,001

```

i	i!	Si	Sum
0	1	3	1
1	1	3,000000000	4,000000000
2	2	4,500000000	8,500000000
3	6	4,500000000	13,000000000
4	24	3,375000000	16,375000000
5	120	2,025000000	18,400000000
6	720	1,012500000	19,412500000
7	5040	0,43392857	19,84642857
8	40320	0,16272321	20,00915179
9	362880	0,05424107	20,06339286
10	3628800	0,01627232	20,07966518
11	39916800	0,00443791	20,08410308
12	479001600	0,00110948	20,08521256
13	1932053504	0,00082520	20,08521256

Отримано з вик. ряду  $\text{EXP}(3)=20,0852125608766$   
Отримано з вик. станд. функції  $\text{EXP}(3)=20,0855369231877$

Рисунок 2.8 – Результати виконання програми Тест1

```

Введіть параметри x,e:
x=3
e=0,00001

```

i	i!	Si	Sum
0	1	3	1
1	1	3,000000000	4,000000000
2	2	4,500000000	8,500000000
3	6	4,500000000	13,000000000
4	24	3,375000000	16,375000000
5	120	2,025000000	18,400000000
6	720	1,012500000	19,412500000
7	5040	0,43392857	19,84642857
8	40320	0,16272321	20,00915179
9	362880	0,05424107	20,06339286
10	3628800	0,01627232	20,07966518
11	39916800	0,00443791	20,08410308
12	479001600	0,00110948	20,08521256
13	1932053504	0,00082520	20,08603776
14	1278945280	0,00373978	20,08977753
15	2004310016	0,00715903	20,09693656
16	2004189184	0,02147837	20,11841493
17	-288522240	-0,44759171	20,11841493

Отримано з вик. ряду  $\text{EXP}(3)=20,1184149308908$   
Отримано з вик. станд. функції  $\text{EXP}(3)=20,0855369231877$

Рисунок 2.9 – Результати виконання програми Тест2

```

Введіть параметри x,e:
x=3
e=0,000001
i      i!      Si      Sum
0      1      3      1
1      1      3,000000000  4,000000000
2      2      4,500000000  8,500000000
3      6      4,500000000  13,000000000
4      24      3,375000000  16,375000000
5      120      2,025000000  18,400000000
6      720      1,012500000  19,412500000
7      5040      0,43392857  19,84642857
8      40320      0,16272321  20,00915179
9      362880      0,05424107  20,06339286
10     3628800      0,01627232  20,07966518
11     39916800      0,00443791  20,08410308
12     479001600      0,00110948  20,08521256
13     6227020800      0,00025603  20,08546859
14     87178291200      0,00005486  20,08552346
15     1307674368000      0,00001097  20,08553443
16     20922789888000      0,00000206  20,08553649
17     355687428096000      0,00000036  20,08553649
Отримано з вик. ряду EXP<3>=20,0855364883791
Отримано з вик. станд. функції EXP<3>=20,0855369231877

```

Рисунок 2.10 – Результати виконання програми (з використанням типу long) Тест3

### Лабораторна робота включає 4 завдання:

- 2.1 Обчислювальна задача (конструкція слідування)
- 2.2 Обчислення виразу (використання математичних функцій)
- 2.3 Конструкція розгалуження

В якості  $f(x)$  використовувати:

$\cos(x)$ , для варіантів 1,4,7,10,13

$x^2$ , для варіантів 2,5,8,11,14

$e^x$ , для варіантів 3,6,9,12,15

### 2.4 Циклічні конструкції

Звіт до лабораторної роботи повинен містити: титульний аркуш (назва та номер ЛР, дані автора), мету роботи, індивідуальні завдання (завдання за варіантом), лістинг програми, тестові запуски, відповіді на контрольні запитання, висновки (див. Додатки А, Б).

### Варіант №1

2.1 Обчисліть площу паралелограма за умови, що відома довжина і ширина.

2.2

$$t = \frac{2\cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right).$$

При  $x=14.26$ ,  $y=-1.22$ ,  $z=3.5 \times 10^{-2}$   $t=0.564849$ .

2.3

$$a = \begin{cases} (f(x) + y)^2 - \sqrt{f(x)y}, & xy > 0 \\ (f(x) + y)^2 + \sqrt{|f(x)y|}, & xy < 0 \\ (f(x) + y)^2 + 1, & xy = 0. \end{cases}$$

2.4 Дано додатні числа А і В ( $A > B$ ). На відрізку довжиною А розміщено максимально можлива кількість відрізків довжиною В (без накладання). Не використовуючи операції множення і ділення, знайти довжину незайнятої частини відрізка А.

## Варіант №2

2.1 Обчисліть об'єм паралелепіпеда, якщо відома довжина, ширина, висота.

2.2

$$u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|}(tg^2 z + 1)^x.$$

При  $x=-4.5$ ,  $y=0.75 \times 10^{-4}$ ,  $z=0.845 \times 10^2$   $u=-55.6848$ .

2.3

$$b = \begin{cases} \ln(f(x)) + (f(x)^2 + y)^3, & x/y > 0 \\ \ln|f(x)/y| + (f(x) + y)^3, & x/y < 0 \\ (f(x)^2 + y)^3, & x = 0 \\ 0, & y = 0. \end{cases}$$

2.4 Дано цілі додатні числа N і K. Використовуючи тільки операції додавання і віднімання, знайти частку від ділення без остачі N на K, а також залишок від цього поділу.

## Варіант №3

2.1 Розрахуйте координати середини відрізка (a, b), якщо відрізки задані координатами своїх кінців.

2.2

$$v = \frac{1 + \sin^2(x + y)}{\left|x - \frac{2y}{1 + x^2 y^2}\right|} x^{|y|} + \cos^2\left(\arctg \frac{1}{z}\right).$$

При  $x=3.74 \times 10^{-2}$ ,  $y=-0.825$ ,  $z=0.16 \times 10^2$ ,  $v=1.0553$ .

2.3

$$c = \begin{cases} f(x + y^2 + \sin(y)), & x - y = 0 \\ (f(x) - y)^2 + \cos(y), & x - y > 0 \\ (y - f(x))^2 + tg(y), & x - y < 0. \end{cases}$$

2.4 Дано ціле число N ( $> 0$ ). Знайти подвійний факторіал N:  $N !! = N \cdot (N-2) \cdot (N-4) \cdot \dots$  (останній співмножник дорівнює 2, якщо N - парне, і 1, якщо N - непарне). Щоб уникнути переповнення типу використовуйте тип long або ulong, а N обмежити зверху значенням 25.

#### Варіант №4

2.1 Обчисліть площу трикутника, якщо відомі довжини його сторін.

2.2

$$w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left( 1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$$

При  $x=0.4 \times 10^4$ ,  $y=-0.875$ ,  $z=-0.475 \times 10^{-3}$   $w=1.9873$ .

2.3

$$d = \begin{cases} (f(x) - y)^3 + \arctg(f(x)), & x > y \\ (y - f(x))^3 + \arctg(f(x)), & y > x \\ (y + f(x))^3 + 0.5, & y = x. \end{cases}$$

2.4 Дано ціле число  $N (> 0)$ . Знайти найменше ціле позитивне число  $K$ , квадрат якого перевершує  $N$ :  $K^2 > N$ . Функцію добування квадратного кореня не використовувати.

#### Варіант №5

2.1 Обчисліть площу трапеції, якщо відомі основи  $a$ ,  $b$  і висота  $h$ .

2.2

$$u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|}(tg^2 z + 1)^x.$$

При  $x=-15.246$ ,  $y=4.642 \times 10^{-2}$ ,  $z=20.001 \times 10^2$   $\alpha=-182.036$ .

2.3

$$e = \begin{cases} i\sqrt{f(x)}, & i - \text{непарне, } x > 0 \\ i/2\sqrt{|f(x)|}, & i - \text{парне, } x < 0 \\ \sqrt{|if(x)|}, & \text{інакше.} \end{cases}$$

2.4 Задано ціле число  $N (> 0)$ , перевірити чи є дане число простим.

#### Варіант №6

2.1 Знайдіть суму, різницю, добуток і частку двох цілих чисел ( $a! = 0$ ,  $b! = 0$ ).

2.2

$$\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})}(\arcsin^2 z - |x - y|).$$

При  $x=16.55 \times 10^{-3}$ ,  $y=-2.75$ ,  $z=0.15$   $\beta=-38.902$ .

2.3

$$g = \begin{cases} e^{f(x)-|b|}, & 0.5 < x < 10 \\ \sqrt{|f(x) + b|}, & 0.1 < x < 0.5 \\ 2f(x), & \text{інакше.} \end{cases}$$

2.4 Дано ціле число  $N (> 1)$ . Знайти найменше ціле число  $K$ , при якому виконується нерівність  $3^K > N$ .

### Варіант №7

2.1 Знайдіть корінь квадратний від суми квадратів двох дійсних чисел.

2.2

$$\gamma = 5 \arctg(x) - \frac{1}{4} \arccos(x) \frac{x+3|x-y|+x^2}{|x-y|z+x^2}.$$

При  $x=0.1722$ ,  $y=6.33$ ,  $z=3.25 \times 10^{-4}$   $\gamma=-172.025$ .

2.3

$$s = \begin{cases} e^{f(x)}, & 1 \leq x \leq 10 \\ \sqrt{|f(x) + 4 * b|}, & 12 \leq x \leq 40 \\ bf(x), & \text{інакше.} \end{cases}$$

2.4 Дано ціле число  $N (> 1)$ . Вивести найменше із цілих чисел  $K$ , для яких сума  $1 + 2 + \dots + K$  буде більше або дорівнює  $N$ , і саму цю суму.

### Варіант №8

2.1 Складіть алгоритм перерахунку ваги з фунтів в кілограми (1 фунт дорівнює 409,5 гр).

2.2

$$\phi = \frac{e^{|x-y|}|x-y|^{x+y}}{\arctg(x) + \arctg(z)} + \sqrt[3]{x^6 + \ln^2 y}.$$

При  $x=-2.235 \times 10^{-2}$ ,  $y=2.23$ ,  $z=15.221$   $\phi=39.374$ .

2.3

$$j = \begin{cases} \sin(5f(x) + 3m|f(x)|), & -1 \leq m \leq x \\ \cos(3f(x) + 5m|f(x)|), & x \leq m \\ (f(x) + m)^2, & x = m. \end{cases}$$

2.4 Початковий внесок в банку дорівнює 1000 грн. Через кожен місяць розмір вкладу збільшується на  $P$  відсотків від наявної суми ( $P$  - дійсне число,  $0 < P < 25$ ). За даним  $P$  визначити, через скільки місяців розмір вкладу перевищить 1100 грн., вивести знайдену кількість місяців  $K$  (ціле число) і підсумковий розмір вкладу  $S$  (дійсне число).

### Варіант №9

2.1 Визначте середнє геометричне та середнє арифметичне 5 додатних дійсних чисел.

2.2

$$\psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}.$$

При  $x=1.825 \times 10^2$ ,  $y=18.225$ ,  $z=-3.298 \times 10^{-2}$   $\psi=1.2131$ .

2.3

$$l = \begin{cases} 2f(x+3p^2), & x \leq |p| \\ |f(x) - p|, & 3 \leq x \leq |p| \\ (f(x) - p)^2, & x = |p|. \end{cases}$$



2.4 Клієнт оформлює депозит в банку і кладе на депозит  $X$  грн, під 12% річних з помісячною капіталізацією відсотків до вкладу. Визначити мінімальну початкову суму  $X$ , що дасть змогу отримати після 2 років накопичення суму в розмірі не менше 25 000 грн.

### Варіант №10

2.1 Складіть алгоритм перетворення унцій в грами.

2.2

$$a = 2^{-x} \sqrt{x + \sqrt[4]{|y|}^3 \sqrt{e^{x-1/\sin z}}}.$$

При  $x=3.981 \times 10^{-2}$ ,  $y=-1.625 \times 10^3$ ,  $z=0.512$   $a=1.26185$ .

2.3

$$k = \begin{cases} \ln(|f(x)| + |q|), & |xq| > 10 \\ e^{f(x)+q}, & |xq| < 10 \\ f(x) + q, & |xq| = 10 \end{cases}$$

2.4 Спортсмен-лижник почав тренування, пробігши в перший день 10 км. кожного наступного дня він збільшував довжину пробігу на  $P$  відсотків від пробігу попереднього дня ( $P$  - дійсне,  $0 < P < 50$ ). За даним  $P$  визначити, після якого дня сумарний пробіг лижника за всі дні перевищить 200 км, і вивести знайдене кількість днів  $K$  (ціле) і сумарний пробіг  $S$  (дійсне число).

### Варіант №11

2.1 Розрахуйте витрати пального на маршрут, якщо задано: відстань та середні витрати пального.

2.2

$$b = y^{\sqrt[3]{|x|}} + \cos^3(y) \frac{|x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{|x-y| + \frac{x}{2}}}.$$

При  $x=6.251$ ,  $y=0.827$ ,  $z=25.001$   $b=0.7121$ .

2.3

$$m = \frac{\max(f(x), y, z)}{\min(f(x), y)} + 5.$$

2.4 Дано ціле число  $N$  ( $> 0$ ). Вивести всі цифри, що входять до складу числа

### Варіант №12

2.1 Обчисліть об'єм циліндра, необхідні параметри визначити самостійно.

2.2

$$c = 2^{(y^x)} + (3^x)^y - \frac{y \left( \operatorname{arctg} z - \frac{\pi}{6} \right)}{|x| + \frac{1}{y^2 + 1}}.$$

При  $x=3.251$ ,  $y=0.325$ ,  $z=0.466 \times 10^{-4}$   $c=4.025$ .

2.3

$$n = \frac{\min(f(x) + y, y - z)}{\max(f(x), y)}.$$

2.4 Дано ціле число  $N (> 0)$ . Використовуючи операції ділення без остачі і взяття залишку від ділення, знайти кількість і суму його цифр.

### Варіант №13

2.1 Визначте вартість за послуги водопостачання, якщо відомо попередні та поточні показники лічильника, а також вартість  $1 \text{ м}^3$  води.

2.2

$$f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \operatorname{tg} z)}.$$

При  $x=17.421$ ,  $y=10.365 \times 10^{-3}$ ,  $z=0.828 \times 10^5$   $f=0.33056$ .

2.3

$$p = \frac{|\min(f(x), y) - \max(y, z)|}{2}.$$

2.4 Дано ціле число  $N (> 0)$ . Використовуючи операції ділення без остачі і взяття залишку від ділення, знайти число, отримане при прочитанні числа  $N$  справа наліво.

### Варіант №14

2.1 Визначити вартість фарби необхідної для фарбування баку циліндричної форми (необхідні параметри визначити самостійно).

2.2

$$g = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{-1/\sin z}.$$

При  $x=12.3 \times 10^{-1}$ ,  $y=15.4$ ,  $z=0.252 \times 10^3$   $g=82.8257$ .

2.3

$$q = \frac{\max(f(x) + y + z, xyz)}{\min(f(x) + y + z, xyz)}.$$

2.4 Дано ціле число  $N (> 0)$ . За допомогою операцій ділення без остачі і взяття залишку від ділення визначити, чи є в записі числа  $N$  цифра «2».

### Варіант №15

2.1 Визначити вартість фарби необхідної для фарбування баку, що має форму паралелепіпеда (необхідні параметри визначити самостійно).

2.2

$$h = \frac{x^{y+1} + e^{y-1}}{1 + x|y - tgz|} \left( 1 + |y - x| \right) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}.$$

При  $x=2.444$ ,  $y=0.869 \times 10^{-2}$ ,  $z=-0.13 \times 10^3$   $h=-0.49871$ .

2.3

$$r = \max(\min(f(x), y), z).$$

2.4 Дано ціле число  $N (> 1)$ . Послідовність чисел Фібоначчі  $F_K$  визначається наступним чином:  $F_1 = 1$ ,  $F_2 = 1$ ,  $F_K = F_{K-2} + F_{K-1}$ ,  $K = 3, 4, \dots$  перевірити, чи є число  $N$  числом Фібоначчі. Якщо є, то вивести True, якщо ні - вивести False.

### Контрольні запитання:

1. Чим відрізняється повна та скорочена форма розгалуження?
2. Що таке цикл?
3. З чого складається цикл?
4. Які типи циклів можна реалізувати на C#?
5. Чим відрізняється оператор циклу do-while від оператора while?

## Лабораторна робота №3

### Тема: Обробка масивів та рядків

**Мета:** вивчити протокол оголошення та ініціалізації одновимірних і багатовимірних масивів у мові C#. Набути навичок в реалізації одновимірних масивів, двовимірних масивів (матриць), рядкових змінних засобами мови C#

### Приклад 1. Робота з матрицями

#### Завдання

Є дійсна квадратна матриця порядку  $n$ , усі елементи якої різні. Знайти максимальні елементи на головній та на побічній діагоналі та поміняти їх місцями.

#### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 3.1.

#### Програмна реалізація

```
namespace Lab3.1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Завдання");
            Console.WriteLine("Є дійсна квадратна матриця
порядку  n\n"+
                                "Знайти максимальні елементи на
гол. та на поб. діагоналі, поміняти їх місцями.");
            Console.WriteLine("Введіть розм. матриці:");
            Console.Write("n=");
            int n =
Convert.ToInt32(Console.ReadLine()); //запит розміру
            // створюємо масив
            int[,] myArr = new int[n, n];
            // об'єкт класу Random для генерації випадкових чисел
            (заповнення масиву)
            Random rnd = new Random();
            // Заповнення та виведення масиву
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    myArr[i, j] = rnd.Next(1, 99);
                    Console.Write("{0}\t", myArr[i, j]);
                }
                Console.WriteLine();
            }
            Console.WriteLine("=====");
        }
    }
}
```

```

        int max1 = myArr[0, 0], max2 = myArr[n - 1, 0]; //
еталони максимумів
        int i1=0,i2=n-1,j2=0; //позиції максимумів
        //прохід по діагоналі
        for (int i = 0; i < n ; i++)
        {
            if (myArr[i,i]>max1) //головна діагональ
            {
                max1 = myArr[i, i];
                i1 = i;
            }
            if (myArr[n - 1 - i, i] > max2) //побічна
діагональ
            {
                max2 = myArr[n - 1 - i, i];
                i2 = n - 1 - i;
                j2 = i;
            }
        }
        //виведення знайдених максимумів
        Console.WriteLine("Макс. ел. головної діагоналі
A[{0},{1}]= {2}", i1, i1, myArr[i1,i1]);
        Console.WriteLine("Макс. ел. побічної діагоналі
A[{0},{1}]= {2}", i2, j2, myArr[i2, j2]);

        Console.WriteLine("\n=====");
        //заміна ел.
        myArr[i2, j2] = max1;
        myArr[i1, i1] = max2;
        //виведення масиву з заміненними максимумами
        Console.WriteLine("Масив з зам. елементами");
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                Console.Write("{0}\t", myArr[i, j]);
            }
            Console.WriteLine();
        }
        Console.ReadLine();
    }
}

```

### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів

Тест1: n=6 Результати роботи програми наведені на рисунку 3.2.

Тест2: n=5 Результати роботи програми наведені на рисунку 3.3.

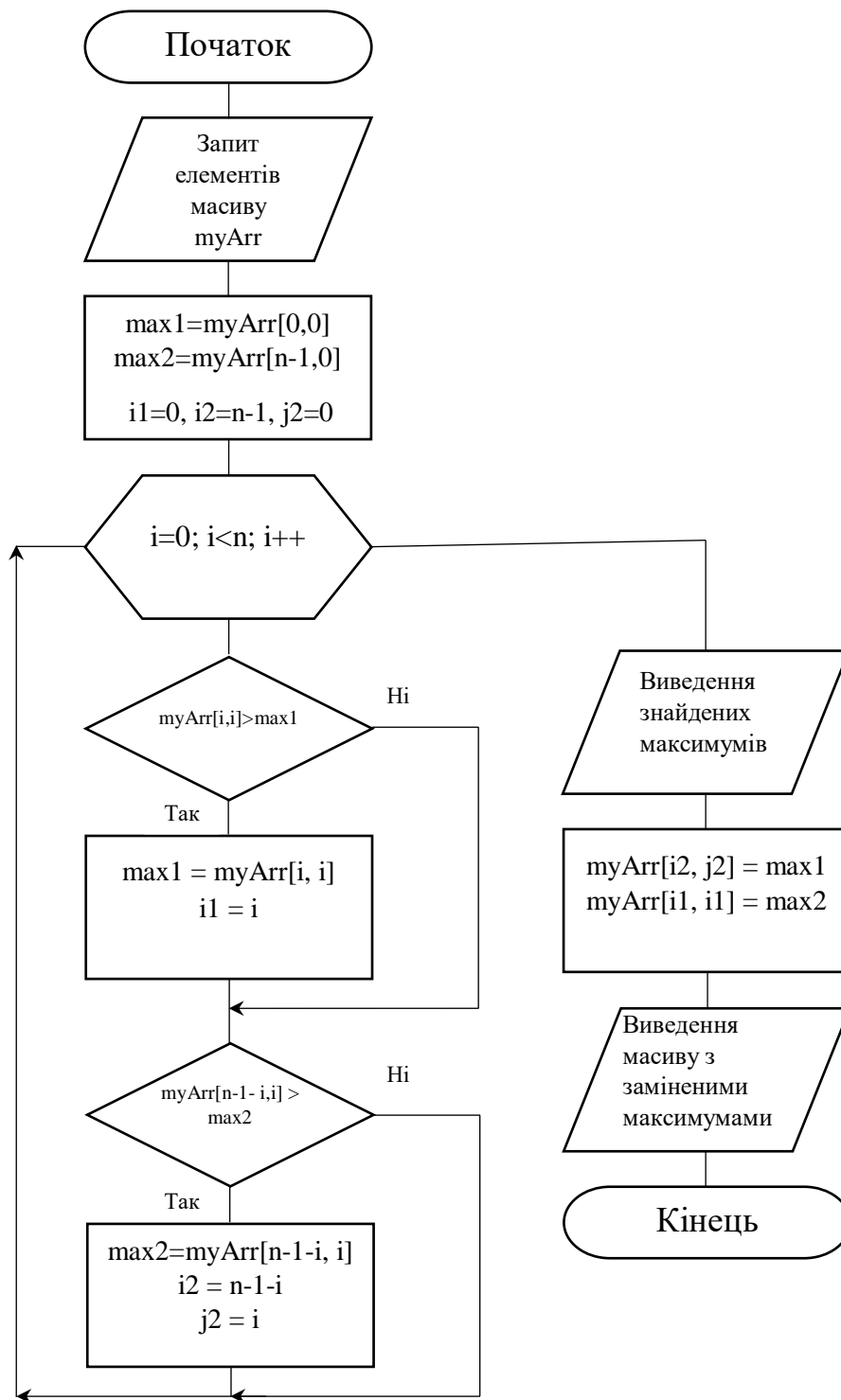


Рисунок 3.1 – Блок-схема заміна максимумів

```

Завдання
Є дійсна квадратна матриця порядку n
Знайти максимальні елементи на гол. та на поб. діаг.
Введіть розм. матриці:
n=6
41      87      33      55      79      59
61      1       24      58      18      3
39      84      66      40      27      51
49      14      16      10      83      97
33      96      30      47      23      42
2       4       60      83      1       78
=====
Макс. ел. головної діагоналі A[5,5]=78
Макс. ел. побічної діагоналі A[4,1]=96
=====
Масив з зам. елементами
41      87      33      55      79      59
61      1       24      58      18      3
39      84      66      40      27      51
49      14      16      10      83      97
33      78      30      47      23      42
2       4       60      83      1       96

```

Рисунок 3.2 – Результати виконання програми Тест1

```

Завдання
Є дійсна квадратна матриця порядку n
Знайти максимальні елементи на гол. та на поб. діагоналі?
Введіть розм. матриці:
n=5
66      47      94      20      98
82      28      19      29      32
47      84      57      50      88
96      36      48      87      68
69      47      95      16      19
=====
Макс. ел. головної діагоналі A[3,3]=87 ✓
Макс. ел. побічної діагоналі A[0,4]=98 ✓
=====
Масив з зам. елементами
66      47      94      20      87
82      28      19      29      32
47      84      57      50      88
96      36      48      98      68
69      47      95      16      19

```

Рисунок 3.3 – Результати виконання програми Тест2

## Приклад 2. Робота з двовимірними масивами

### Завдання

Задано квадратну матрицю дійсних чисел, розмірність матриці 5x5.

Отримати індекси всіх елементів з найбільшим значенням

### Словесний опис алгоритму

Фактично задача зводиться до пошуку максимуму в масиві. Але задача ускладнюється тим, що максимумів може бути декілька (тобто декілька елементів мають однакове, більше від інших значення).

Пропонується наступний алгоритм:

1) Шукаємо максимум за класичним алгоритмом, методом порівняння з еталоном.

2) Якщо зустрічаємо елемент рівний поточному максимуму зберігаємо його індекси (зручно використати List)

3) Якщо знаходимо більше значення – очищуємо раніше накопичені елементи (попередні максимуми)

4) Повторюємо пункти 1-3 для всіх елементів матриці

В список необхідно додавати два індекси. Тут можна запропонувати багато різних варіантів: використати структуру, клас або інші види конструкцій. Але враховуючи, що розмірність матриці 5x5 (індекси є однозначними числами) можна просто запакувати два індекси в одне ціле двоцифрове число: кількість десятків – індекс рядка, кількість одиниць – індекс стовпчика.

Так як інтерес являє сам алгоритм обробки, а не конкретні значення елементів, масив можна заповнити випадковими числами.

### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 3.4.

### Програмна реалізація

```
namespace Lab3.2
{
    class Program
    {
        static void Main(string[] args)
        {
            double[,] myArr = new double [5,5];
            Random rnd = new Random();
            List<int> mL = new List<int>();
            //заповнення та виведення елементів
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    myArr[i, j] = rnd.Next(1, 20);
                    Console.Write(" {0}",
String.Format("{0,10:0.0}", myArr[i, j]));
                    //String.Format("{0,10:0.0}",myArr[i, j]);
                }
                Console.WriteLine();
            }
            //пошук максимального елемента
            mL.Clear();
            double max = myArr[0, 0];
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    //якщо значення елемента рівне
максимальному
                    if (Math.Abs(myArr[i, j]-max)<0.0000001)
```



```

        {
            mL.Add((i + 1) * 10 + (j + 1));
        }
        //якщо знайшли новий макс.
        if (myArr[i, j] > max)
        {
            max = myArr[i, j]; //змінюємо максимум
            mL.Clear(); //очищуємо список
            mL.Add((i + 1) * 10 + (j + 1));
        }
        //додаємо індекси нового максимуму
    }
}

Console.WriteLine("Індекси макс. ел.");
foreach (int tmp in mL)
{
    Console.Write(" [{0},{1}] ", tmp / 10 - 1, tmp % 10 - 1);
}

Console.ReadLine();
}
}
}

```

### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів.

Тест1: Результати роботи програми наведені на рисунку 3.6.

Тест2: Результати роботи програми наведені на рисунку 3.7.

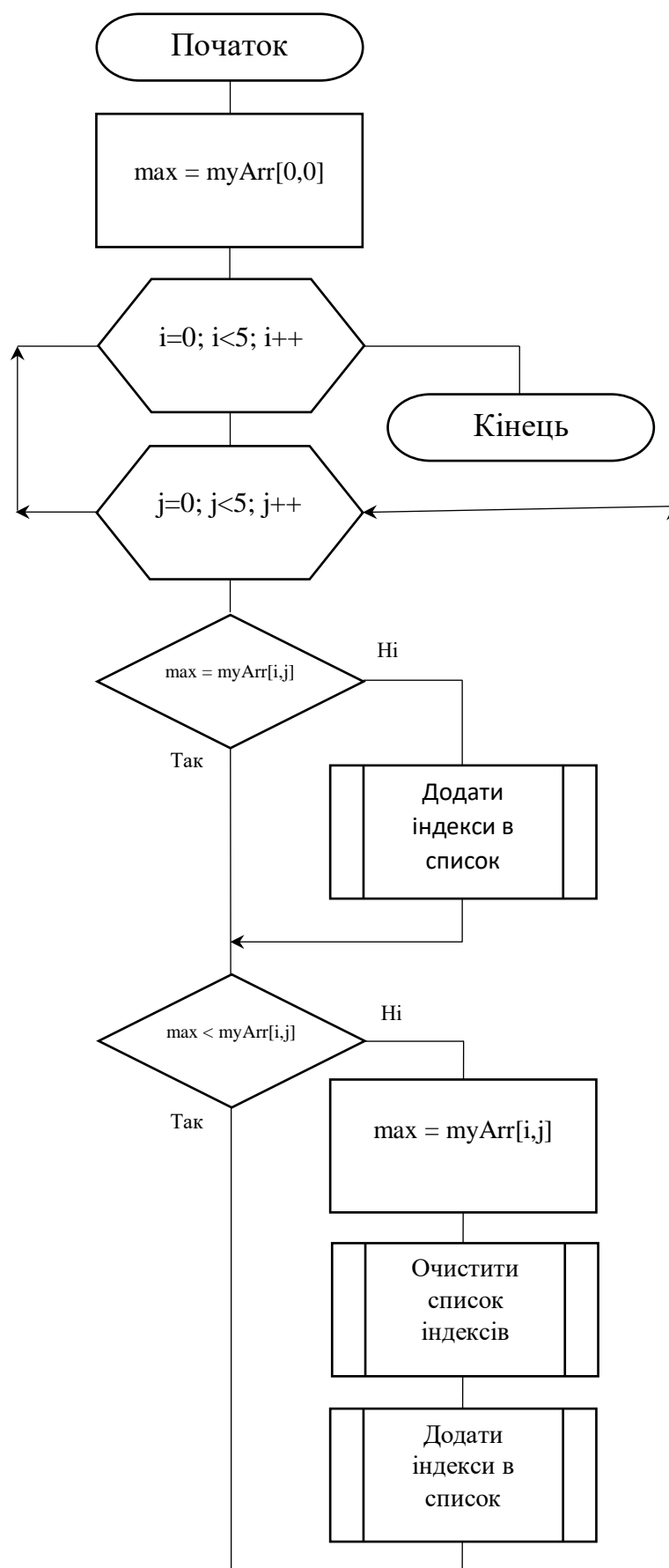


Рисунок 3.4 – Блок-схема алгоритму

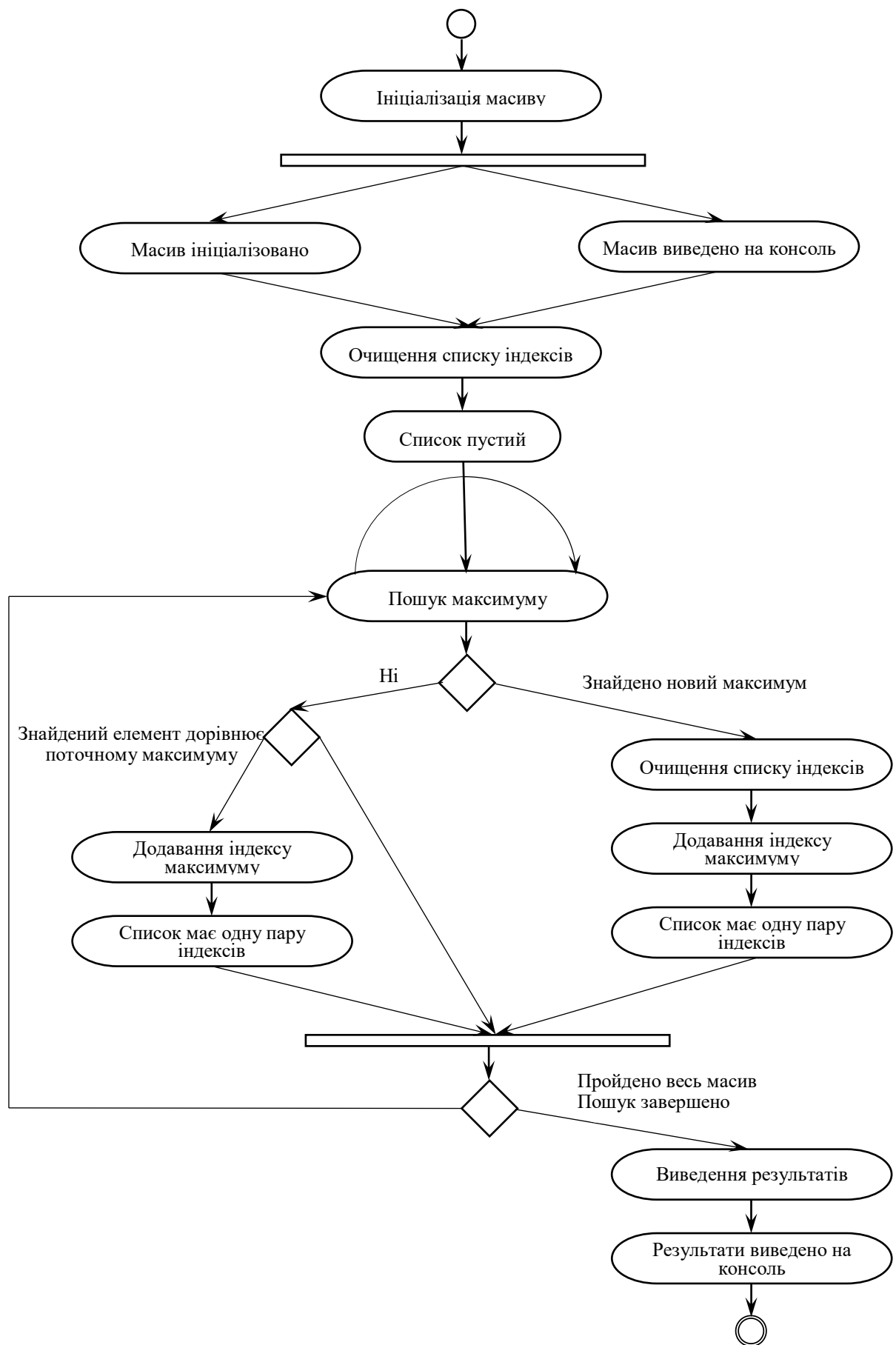


Рисунок 3.5 – Діаграма дій алгоритму

8,0	2,0	5,0	1,0	13,0
18,0	8,0	19,0 ✓	5,0	2,0
6,0	13,0	4,0	15,0	5,0
4,0	12,0	19,0 ✓	8,0	11,0
8,0	18,0	10,0	12,0	1,0
Індекси макс. ел.:				
[1,2]	[3,2]			

Рисунок 3.6 – Результати виконання програми Тест 1

7,0	3,0	13,0	19,0 ✓	3,0
17,0	13,0	16,0	6,0	13,0
15,0	6,0	4,0	17,0	7,0
15,0	4,0	3,0	11,0	8,0
19,0 ✓	5,0	19,0 ✓	12,0	10,0
Індекси макс. ел.:				
[0,3]	[4,0]	[4,2]		

Рисунок 3.7 – Результати виконання програми Тест 2

### Приклад 3. Рядок string

#### Завдання

З клавіатури увести фразу. Перевірити, чи утворюють символи цієї фрази паліндром (текст, симетричний відносно середини). Пропуски, розділові знаки і різницю в регістрах – ігнорувати.

#### Блок-схема алгоритму

Блок-схему алгоритму наведено на рисунку 3.8.

#### Програмна реалізація

```
namespace Lab3.3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Завдання");
            Console.WriteLine("З клавіатури увести фразу.
Перевірити, чи фраза паліндромом");
            Console.WriteLine("Введіть рядок:");
            Console.Write("St:");
            //введення рядка
            string s = Console.ReadLine();
            //регулярний вираз з використанням методу ЗАМІНА
(Replace)
            //Замінити всі символи, що не є літерами на пустий
символ ""
            s=Regex.Replace(s, "[^a-zA-Z]", "");
```

```

        s = s.ToLower(); //переводимо всі символи в нижній
регістр
        string rev_s = new
string(s.Reverse().ToArray()); // виконуємо реверс масиву, і
повертаємо знову в тип string
        //виведення прямого і інверсного рядків
        Console.WriteLine("{0}", s);
        Console.WriteLine("{0}", rev_s);
        //порівняння рядків прямого і інверсного
        if (s == rev_s)
        {
            Console.WriteLine("Фраза є паліндромом");
        }
        else
        {
            Console.WriteLine("Фраза не є паліндромом");
        }

        Console.ReadLine();
    }
}

```

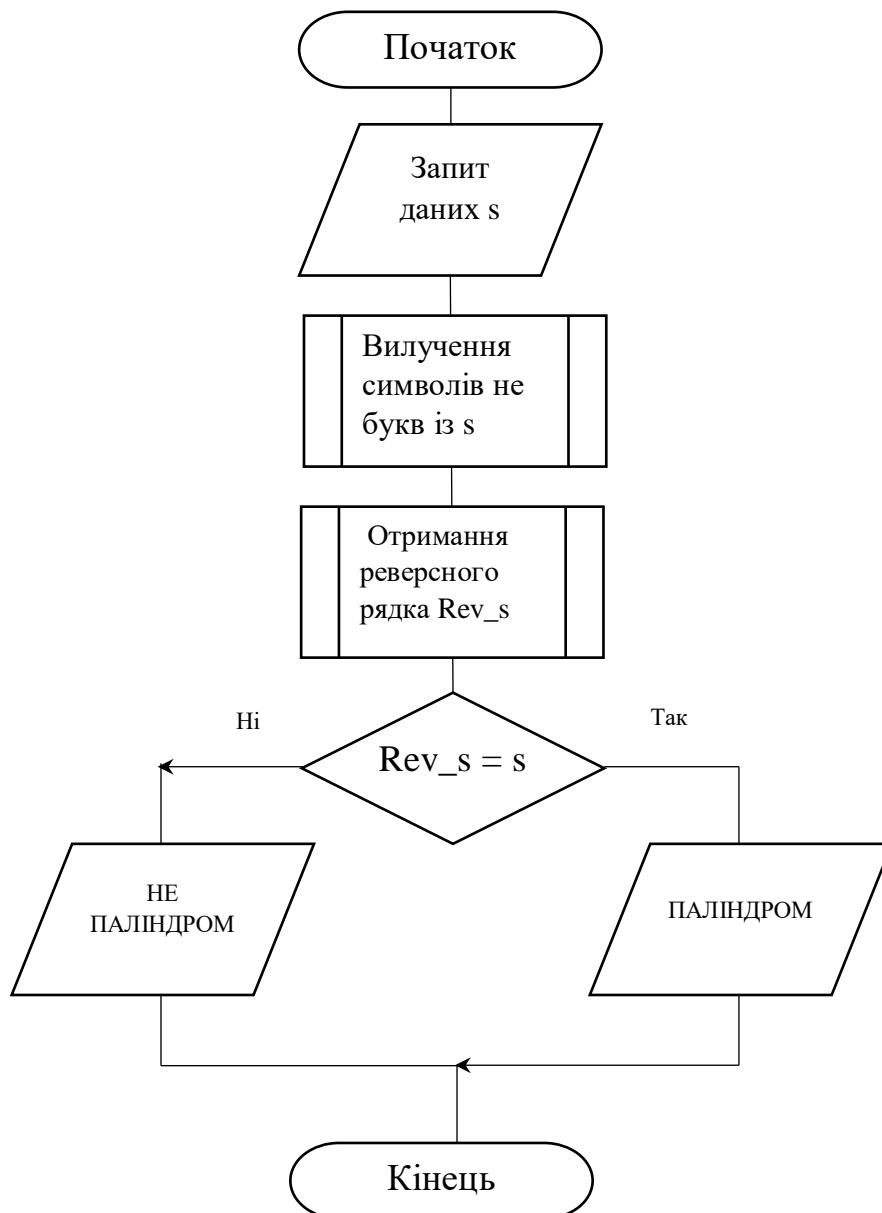


Рисунок 3.8 – Блок-схема «ПАЛІНДРОМ»

### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів.

Тест1 (паліндром) : Результати роботи програми наведені на рисунку 3.9.

```

Завдання
З клавіатури увести фразу. Перевірити, чи фраза паліндромом
Введіть рядок:
St:asd4578 DsA.,
asddsa
asddsa
Фраза є паліндромом

```

Рисунок 3.9 – Результати виконання програми Тест1

Тест2 (паліндром) : Результати роботи програми наведені на рисунку 3.10.

```

Завдання
З клавіатури увести фразу. Перевірити, чи фраза паліндромом
Введіть рядок:
St:a roza upala na laru azora
arozauralanalaruazora
arozauralanalaruazora
Фраза є паліндромом

```

Рисунок 3.10 – Результати виконання програми Тест2

Тест3 (не паліндром) : Результати роботи програми наведені на рисунку 3.11.

```

Завдання
З клавіатури увести фразу. Перевірити, чи фраза паліндромом
Введіть рядок:
St:kadjsnj676JDJedfe23, sdb
kadjsnjjdjedfesdb
bdsefdejdjjnsjda
Фраза не є паліндромом

```

Рисунок 3.11 – Результати виконання програми Тест3

### Лабораторна робота включає 3 завдання:

3.1 Одновимірний масив<sup>3</sup>

3.2 Двовимірний масив (матриця)<sup>4</sup>

3.3 Робота з рядковими змінними

Звіт до лабораторної роботи повинен містити: титульний аркуш (назва та номер ЛР, дані автора), мету роботи, індивідуальні завдання (завдання за варіантом), лістинг програми, тестові запуски, відповіді на контрольні запитання, висновки (див. Додатки А, Б). В завданні 3.2 окрім лістингу та тестових запусків розробити блок-схему алгоритму.

### Варіант №1

3.1 У масиві з 20 цілих чисел знайти найбільший елемент і поміняти його місцями з першим елементом.

3.2 Задано цілочисельну матрицю N (10,10). Обчислити суму і кількість позитивних елементів матриці, що знаходяться над головною діагоналлю.

<sup>3</sup> Для всіх варіантів масиви заповнюються випадковими числами

<sup>4</sup> Масиви заповнюються випадковими числами, якщо в завданні не вказано конкретних значень елементів

3.3 Дано рядок, що складається з груп нулів і одиниць. Порахувати кількість нулів і одиниць.

### **Варіант №2**

3.1 В масиві з 10 цілих чисел знайти найменший елемент і поміняти його місцями з останнім елементом.

3.2 Задано цілочисельну матрицю N (10,10). Записати на місце від'ємних елементів матриці нулі і вивести її на екран в загальноприйнятому вигляді (у вигляді таблиці).

3.3 Порахувати в рядку кількість слів.

### **Варіант №3**

3.1 У масиві з 20 цілих чисел знайти найбільший елемент і поміняти його місцями з першим елементом.

3.2 Задано цілочисельну матрицю N (10,10). Записати на місце від'ємних елементів матриці нулі, а на місце позитивних - одиниці. Вивести на екран нижню трикутну матрицю в загальноприйнятому вигляді.

3.3 Знайти кількість знаків пунктуації в заданому рядку.

### **Варіант №4**

3.1 У масиві з 25 дійсних чисел знайти найменший елемент і поміняти його місцями з першим елементом.

3.2 Задано цілочисельну матрицю N (10,10). Знайти в кожному рядку матриці максимальний і мінімальний елементи і помістити їх на місце першого і останнього елемента відповідного рядка. Матрицю вивести в загальноприйнятому вигляді.

3.3 Дано рядок символів. Вивести на екран цифри, що містяться в рядку.

### **Варіант №5**

3.1 Дано масив X, що містить 16 елементів. Обчислити і вивести значення

$$d_i = \frac{e^{x_i} + 2e^{-x_i}}{\sqrt{5 + \sin x_i}}, i, \text{ підрахувати кількість } d_i > 0.1.$$

3.2 Задано цілочисельну матрицю N (10,10). Для цілочисельної матриці знайти для кожного рядка число елементів, кратних п'яти, і найбільший з отриманих результатів.

3.3 Дано рядок символів, що складається з довільних десяткових цифр, розділених пробілами. Вивести кількість парних чисел в цьому рядку.

### **Варіант №6**

3.1 Дано масив Y, що містить 25 елементів. Записати в масив R і вивести значення елементів, які обчислюють за формулою,

$$r_i = \frac{5y_i + \cos^2 y_i}{2.35}, i = 1, 2, \dots, 25.$$



3.2 Задано цілочисельну матрицю  $N$  (10,10). З позитивних елементів матриці  $N$  сформувати матрицю  $M$  (10,  $K_{\max}$ ), розташовуючи їх в рядках матриці поспіль, де  $K_{\max}$  - максимальне число позитивних елементів рядка матриці  $N$ . Записати нулі на місце відсутніх елементів. Вивести обидві матриці в загальноприйнятому вигляді.

3.3 Дано рядок символів. Вивести на екран процентне співвідношення голосних та приголосних літер.

### **Варіант №7**

3.1 Дано масив  $F$ , що містить 18 елементів. Обчислити і вивести елементи нового масиву  $p_i = 0.13f_i^3 - 2.5f_i + 8$ . Окремо вивести негативні елементи масиву  $P$ .

3.2 Задано цілочисельну матрицю  $N$  (10,10). Знайти в кожному рядку найбільший елемент і поміняти його місцями з елементом головної діагоналі. Вивести матрицю в загальноприйнятому вигляді.

3.3 Дано рядок символів. Вивести на екран тільки символи верхнього регістру, що входять до цього рядка.

### **Варіант №8**

3.1 Обчислити і вивести елементи масиву  $Z$ , де  $z_i = i^2 + 1$ , якщо  $i$  - непарне, і  $z_i = 2i - 1$ , якщо  $i$  - парне. Сформувати і вивести масив  $D$ :  $d_i = 2.5z_i$ , якщо  $z_i > 2.5$  і  $d_i = z_i / 2.5$ , якщо  $z_i \leq 2.5$ .

3.2 Задано цілочисельну матрицю  $N$  (10,10). Знайти найбільший і найменший елементи матриці і поміняти їх місцями. Матрицю вивести в загальноприйнятому вигляді.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. У кожному слові замінити першу букву на прописну.

### **Варіант №9**

3.1 У масиві  $R$ , що містить 25 елементів, замінити значення негативних елементів квадратами значень, значення позитивних збільшити на 7, а нульові значення залишити без зміни. Вивести масив  $R$ .

3.2 Сформувати цілочисельну матрицю  $N$  (24,7). Сформувати нову матрицю  $M$ , обчисливши середнє арифметичне значення елементів кожного рядка  $N$  і записати його в 8-й стовпець матриці  $M$ , а також середнє арифметичне кожного стовпчика  $N$  і записати його в 25-й рядок матриці  $M$ . Матрицю вивести в загальноприйнятому вигляді.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. Видалити першу букву в кожному слові.

### **Варіант №10**

3.1 Дано масив  $A$  цілих чисел, що містить 30 елементів. Обчислити і вивести суму тих елементів, які кратні 5.

3.2 Задано цілочисельну матрицю  $N(10,10)$ . Знайти рядок з найбільшою і найменшою сумою елементів. Вивести на екран знайдені рядки і суми їх елементів.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. Поміняти місцями  $i$ - та  $j$ -ю букви в кожному слові.

### **Варіант №11**

3.1 Дано масив  $A$  цілих чисел, що містить 30 елементів. Обчислити і вивести суму тих елементів, що є непарними і від'ємними.

3.2 Сформувати цілочисельну матрицю  $N(24,12)$ . Знайти рядки, що мають максимальну кількість співпадіння значень.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. Поміняти місцями першу і останню букви кожного слова.

### **Варіант №12**

3.1 Дано масив  $A$  цілих чисел, що містить 30 елементів. Обчислити і вивести суму тих елементів, які задовольняють умові  $|a_i| < i^2$ .

3.2 Дана матриця  $N(10,10)$ , що містить значення 0,1,2,3. Визначити номер рядка з найбільшою кількістю нулів.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. Замінити всі букви латинського алфавіту на знак «+».

### **Варіант №13**

3.1 Дано масив  $A$  цілих чисел, що містить 30 елементів. Обчислити і вивести кількість і суму тих елементів, які діляться на 5 і не діляться на 7.

3.2 Дана матриця  $N(10,10)$ , що містить значення 0,1. Визначити номери рядків матриці, в яких значення елементів чергуються.

3.3 Дано рядок символів, що містить деякий текст записаний символами кирилиці. Замінити всі великі літери «А» на символ «\*».

### **Варіант №14**

3.1 Дано масив  $A$  дійсних чисел, що містить 25 елементів. Обчислити і вивести число негативних елементів і число членів, що належать відріzkу  $[1,12]$ .

3.2 Дана матриця  $N(11,11)$ . Обійти матрицю по спіралі починаючи від центрального елементу  $N[5][5]$ , проти руху часової стрілки.

3.3 Дано рядок символів, що містить деякий текст. Розробити програму, яка визначає, чи є даний текст паліндромом, тобто читається він зліва направо так само, як і справа наліво без урахування регістру(наприклад, «А роза упала на лапу Азора»).

### **Варіант №15**

3.1 Дано масив  $Z$  цілих чисел, що містить 35 елементів. Обчислити і вивести  $R = S + P$ , де  $S$  - сума парних елементів, менших за 3,  $P$  - добуток непарних елементів, більших за 1.

3.2 Визначити найменший елемент кожного парного рядка матриці  $N$  (24,12), та знайти їх суму.

3.3 Дано рядок символів, що складається з довільного тексту англійською мовою, слова розділені пробілами. Сформувати новий рядок, що складається з чисел довжин слів в заданій стрічці. Визначити найдовше слово.

### **Контрольні запитання:**

1. Яку функція виконують і як використовуються індекси елементів масиву?
2. Що визначає розмірність масиву?
3. Як використовується команда `new`?
4. Чи можна при описі масиву не вказувати кількість елементів, яким чином в цьому випадку буде визначено розмір масиву?
5. В чому переваги типу `string` порівняно з масивом символів?
6. Перелічить основні методи класу `String`?

## Лабораторна робота №4

### Тема: Алгоритми сортування та пошуку

**Мета:** закріпити теоретичні знання з теми алгоритмів сортування та пошуку. Реалізувати бінарний пошук в відсортованому масиві засобами мови C#

### Приклад обробки масивів

#### Завдання

В матриці  $C$  кількість стовпчиків у кожному рядку є випадковим натуральним числом з інтервалу  $[a; b]$ , але загальна кількість елементів є квадратом натурального числа. Переписати всі елементи з матриці  $C$  в одновимірний масив  $F$ , відсортувати його за зростанням, після чого переписати його елементи у квадратну матрицю  $Q$  по рядках.

#### Аналіз

З умови задачі не зрозумілим є момент пов'язаний з розмірністю та значеннями початкової матриці  $C$ .

1. Так як ніяких додаткових вимог не встановлено – елементи будуть генеруватись випадковим чином в діапазоні  $(0..20)$  (останнє обмеження лише для зручності виведення).

2. Не вказано кількість рядків, кількість стовбців випадкове число з інтервалу  $[a; b]$ , але при цьому загальна кількість має бути повним квадратом (теж невідомого числа)... Повністю цю умову виконати неможливо, бо сума випадкових чисел (кількість елементів в рядку) не завжди (точніше – дуже рідко) буде повним квадратом. Відповідно необхідно встановлювати не випадкову кількість елементів принаймні для частини рядків. Натуральне число ( $N$ ), повним квадратом якого є кількість елементів, буде вводиться в програмі. Кількість рядків буде визначатись динамічно, рядки додаються поки кількість згенерованих елементів  $\leq N^2 - (a+b)$ . Потім ще один з кількістю не вистачаючих до повного квадрату елементів.

Але в такому випадку доведеться на кожній ітерації додавання рядка змінювати розмірність масиву. Для цього є метод `Resize` але (як пишуть форуми і документація) це дуже не ефективний метод. Для такої ситуації необхідно (краще) використовувати динамічні структури (наприклад `List`). Можливо умова трактується не правильно, але в такому трактуванні знаходиться лише такий вихід.

3. Інші пункти умови абсолютно зрозумілі: Розмірність матриці  $F$  –  $1 \times N^2$   
Розмірність матриці  $Q$  –  $N \times N$

### Програмна реалізація

```
/*  
 * В матриці C кількість стовпчиків у кожному рядку є  
 випадковим натуральним числом з інтервалу [a; b],  
 * але загальна кількість елементів є квадратом натурального  
 числа.
```

\* Переписати всі елементи з матриці С в одновимірний масив F, відсортувати його за зростанням,  
 \* після чого переписати його елементи у квадратну матрицю Q по рядках.  
 \*/

```
namespace Array_Lab_4
{
    class Program
    {
        static void PrintToolzArray(int[][] ToolzArray)
//Виведення зубчастого масиву.
        {
            foreach (var row in ToolzArray)
            {
                Console.WriteLine(string.Join(", ", row));
            }
        }
        static void Main(string[] args)
        {
            int a=0, b=0, N = 0;
            int rc = 0, cc = 0;
            int el = 0, count_el=0;
            //Запит даних
            Console.Write("Введіть N, квадрат якого визначає
кількість ел:");
            N = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Введіть межі кіл. елементів
рядка зубчастого масиву");
            Console.Write("(a>=1) a=");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("(b<=N*N) b=");
            b = Convert.ToInt32(Console.ReadLine());
            //Створення екз. класу Random для генерації
випадкових чисел
            Random rnd = new Random();
            int[][] C = new int[1][]; //опис зубчастого масиву
            //спочатку масив має лише 1 рядок
            rc = 0;
            while (count_el < (N * N - (a + b)))
            {
                cc = rnd.Next(a, b); //кількість стовбців
(елементів рядка)
                C[rc] = new int[cc]; //створення рядка
                //заповнення рядка
                for (int i = 0; i < cc; i++)
                {
                    C[rc][i] = rnd.Next(20);
                }
                count_el += cc; //підрахунок ел.
                //Якщо ел. не вистачає збільшуємо кількість
рядків
                if (count_el <= (N * N - (a + b)))
                {

```

```

        rc++;
        Array.Resize(ref C, rc+1); //додаємо рядок
    }
    // останній компенсуючий рядок, щоб кількість
відповідала повному квадрату
    rc++;
    Array.Resize(ref C, rc + 1);
    cc = N * N -count_el; // вираховуємо скільки не
вистачило

    count_el += cc;
    C[rc] = new int[cc];
    for (int i = 0; i < cc; i++)
    {
        C[rc][i] = rnd.Next(20);
    }
    //виведення зубчастого масиву
    Console.WriteLine("Зубчастий масив:");
    PrintToolzArray(C);

Console.WriteLine("=====");
    Console.WriteLine("Кількість ел: {0}", count_el);
    //Робота з лінійним масивом та двомірним масивом
    int[] F = new int[N * N]; //розмірність лінійного
масиву вже задана (кількість ел. зубчастого)
    int[, ] Q = new int[N, N]; //розмірність двомірного
масиву теж задана
    int ind=0;
    Console.WriteLine("Лінійний масив:");
    foreach(int[] row in C)
    {
        foreach(int number in row)
        {
            F[ind] = number;
            Console.Write("{0} ", F[ind]); //виведення
невідсортованих ел.

            ind++;
        }
    }
    Console.WriteLine();
    Array.Sort(F); //Сортування
    //Виведення після сортування
    Console.WriteLine("Лінійний масив
(відсортований):");
    int i1 = 0, i2 = 0;
    // разом з виводом лінійного відсортованого масиву
будемо копіювати його в двомірний масив
    foreach (int number in F)
    {
        Console.Write("{0} ", number);
        Q[i1, i2] = number;
        //керуємо індексами
        i2++;
    }

```

```

        if (i2 > N - 1)
        {
            i1++;
            i2 = 0;
        }
    }
    Console.WriteLine("\n=====");
    //Виведення двомірного масиву
    Console.WriteLine("Двомірний масив:");
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            Console.Write("{0,2} ", Q[i,j]);
        }
        Console.WriteLine();
    }
    Console.WriteLine("=====");
    Console.ReadLine();
}
}
}

```

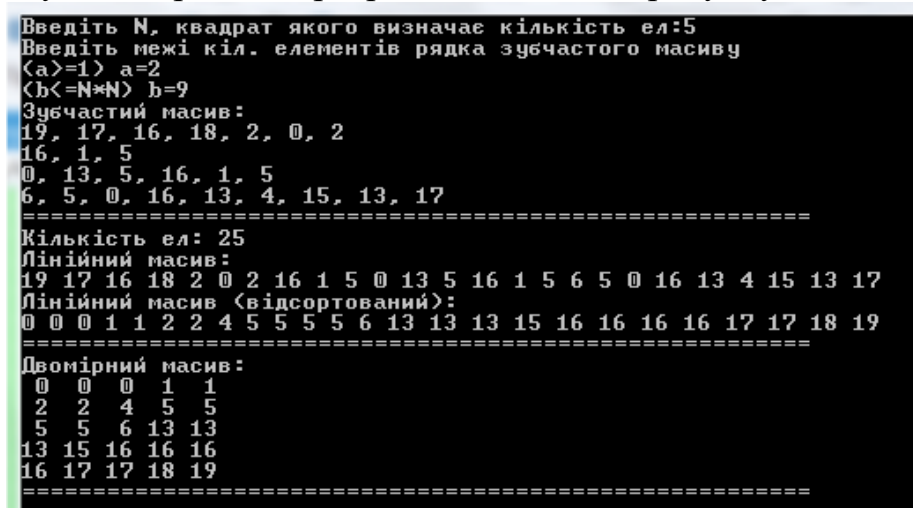
### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів

Тест1: Результати роботи програми наведені на рисунку 4.1.

Тест2: Результати роботи програми наведені на рисунку 4.2.

Тест3: Результати роботи програми наведені на рисунку 4.3.



```

Введіть N, квадрат якого визначає кількість ел:5
Введіть межі кіл. елементів рядка зубчастого масиву
(a>=1) a=2
(b<=N*N) b=9
Зубчастий масив:
19, 17, 16, 18, 2, 0, 2
16, 1, 5
0, 13, 5, 16, 1, 5
6, 5, 0, 16, 13, 4, 15, 13, 17
=====
Кількість ел: 25
Лінійний масив:
19 17 16 18 2 0 2 16 1 5 0 13 5 16 1 5 6 5 0 16 13 4 15 13 17
Лінійний масив (відсортований):
0 0 0 1 1 2 2 4 5 5 5 5 6 13 13 13 15 16 16 16 17 17 18 19
=====
Двомірний масив:
0 0 0 1 1
2 2 4 5 5
5 5 6 13 13
13 15 16 16 16
16 17 17 18 19
=====

```

Рисунок 4.1 – Результати виконання програми Тест1

```

Введіть N, квадрат якого визначає кількість ел:8
Введіть межі кіл. елементів рядка зубчастого масиву
(a>=1) a=3
(b<=N*N) b=18
Зубчастий масив:
3, 6, 1, 17, 13, 16, 0, 16, 11
19, 14, 5, 4, 1, 13, 11, 0, 13, 3, 11, 16, 3, 5, 7
5, 5, 6, 14, 16
15, 12, 2, 11, 1, 3, 16, 2, 4
19, 11, 10, 5, 7, 9, 14, 12
11, 11, 4, 10, 18, 7, 3, 16, 9, 9, 10, 14, 9, 16, 0, 10, 18, 12
=====
Кількість ел: 64
Лінійний масив:
3 6 1 17 13 16 0 16 11 19 14 5 4 1 13 11 0 13 3 11 16 3 5 7 5 5 6 14 16 15 12 2
11 1 3 16 2 4 19 11 10 5 7 9 14 12 11 11 4 10 18 7 3 16 9 9 10 14 9 16 0 10 18 1
2
Лінійний масив (відсортований):
0 0 0 1 1 1 2 2 3 3 3 3 3 4 4 4 5 5 5 5 5 6 6 7 7 7 9 9 9 9 10 10 10 10 11 11 11
11 11 11 11 12 12 12 13 13 13 14 14 14 14 15 16 16 16 16 16 16 16 16 17 18 18 19 1
9
=====
Двомірний масив:
0 0 0 1 1 1 2 2
3 3 3 3 3 4 4 4
5 5 5 5 5 6 6 7
7 7 9 9 9 9 10 10
10 10 11 11 11 11 11 11
11 12 12 12 13 13 13 14
14 14 14 15 16 16 16 16
16 16 16 17 18 18 19 19
=====

```

Рисунок 4.2 – Результати виконання програми Тест2

```

Введіть N, квадрат якого визначає кількість ел:4
Введіть межі кіл. елементів рядка зубчастого масиву
(a>=1) a=2
(b<=N*N) b=6
Зубчастий масив:
11, 18, 5, 16, 5
17, 8
5, 18
0, 11, 16, 14, 3, 3, 11
=====
Кількість ел: 16
Лінійний масив:
11 18 5 16 5 17 8 5 18 0 11 16 14 3 3 11
Лінійний масив (відсортований):
0 3 3 5 5 5 8 11 11 11 14 16 16 17 18 18
=====
Двомірний масив:
0 3 3 5
5 5 8 11
11 11 14 16
16 17 18 18
=====

```

Рисунок 4.3 – Результати виконання програми Тест3

### Лабораторна робота включає 2 завдання:

4.1 Реалізувати алгоритм сортування за варіантом, порівняти його з швидким алгоритмом сортування (кількість порівнянь, кількість перестановок, швидкість роботи в мс.). Провести порівняльний експеримент для вибраних



алгоритмів. В якості вхідного набору даних використати масив цілих чисел розмірності: 1000, 10000, 50000, 100000 елементів.

4.2 Реалізувати алгоритм бінарного пошуку, виконати алгоритм для впорядкованого масиву, знайти середню кількість виконаних порівнянь алгоритму бінарного пошуку (запустивши алгоритм 5-7 разів для різних вхідних даних).

Заповнити звіт. Звіт повинне містити: титульний аркуш (назва та номер ЛР, дані автора), мету роботи, індивідуальні завдання (завдання за варіантом), блок-схеми алгоритмів сортування та пошуку, таблицю з результатами отриманими в ході порівняння, графік залежності кількості порівнянь/перестановок від кількості елементів, лістинг програмного коду, відповіді на контрольні запитання, висновки (див. Додатки А, Б).

### **Варіанти вибору алгоритму сортування:**

Варіанти	Алгоритм
1, 5, 9, 13	Бульбашковий алгоритм
2, 6, 10, 14	Алгоритм простої вставки
3, 7, 11, 15	Алгоритм МінМакс
4, 8, 12	Шейкерний алгоритм

### **Контрольні запитання:**

1. Що таке алгоритм?
2. Які властивості можуть мати алгоритми?
3. Які є види алгоритмів?
4. Що таке блок-схема?
5. Які алгоритми сортування ви знаєте і які вам доводилось реалізовувати?
6. Які алгоритми пошуку ви знаєте і які вам доводилось реалізовувати?

## Лабораторна робота №5

### Тема: Основи об'єктного програмування

**Мета:** закріпити теоретичні знання з теми об'єктно-орієнтованого підходу до програмування. Реалізувати ієрархію класів засобами мови C#

#### Приклад 1

##### Завдання

Створити клас прямокутників `Rectangle`, які характеризуються довжинами своїх сторін. Для ініціалізації об'єктів даного класу розробити конструктор, а також віртуальну функцію обчислення площі фігури. Побудувати клас прямокутних паралелепіпедів `Parallelepiped`, який успадковує клас `Rectangle` і характеризується довжинами своїх сторін. Перевантажити віртуальну функцію на випадок обчислення об'єму паралелепіпеда.

Доцільним є створення в батьківському класі двох віртуальних функцій:

```
virtual double GetArea // буде перевизначена для спадкоємця
Parallelepiped
{
    return a*b;
}
virtual double GetVolume // для прямокутника об'єм дорівнює 0;
{
    return 0;
}
```

#### Програмна реалізація

```
namespace Pr5_Rectangle
{
    class Program
    {
        class MRectanle
        {
            protected double a;
            protected double b;
            public MRectanle(double fa, double fb)
            {
                a = fa;
                b = fb;
            }
            public virtual double GetArea()
            {
                return a * b;
            }
            public virtual double GetVolume()
            {
                return 0;
            }
        }
    }
}
```

```

    }
    class Parallelepiped : MRectanle// успадковуємо клас
MRectanle
    {
        private double c;
        public Parallelepiped(double fa, double fb, double
fc): base(fa, fb)// викликаємо конструктор батька
        {
            c = fc;
        }
        //Перевизначаємо функції
        public override double GetArea()
        {

            return 2*(a*b+a*c+c*b);
        }
        public override double GetVolume()
        {
            return a*b*c;
        }
    }
    static void Main(string[] args)
    {
        MRectanle r1 = new MRectanle (3, 4);
        Console.WriteLine("Прямокутник зі сторонами a=3 і
b=4");

        Console.WriteLine("Об'єм V={0}", r1.GetVolume());
        Console.WriteLine("Площа S={0}", r1.GetArea());
        Parallelepiped pr1 = new Parallelepiped (3, 4, 2);
        Console.WriteLine("Паралелепіпед зі сторонами a=3,
b=4, c=2");

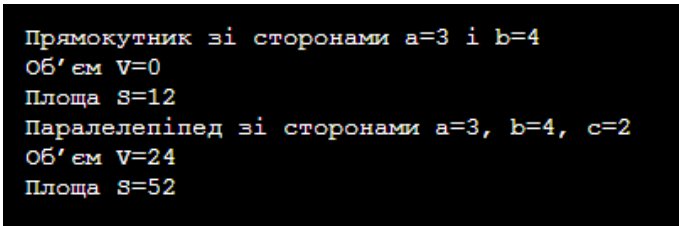
        Console.WriteLine("Об'єм V={0}", pr1.GetVolume());
        Console.WriteLine("Площа S={0}", pr1.GetArea());
        Console.ReadLine();

    }
}

```

### Результати виконання програми

Працездатність програми перевірено із використанням контрольних прикладів, які наведені на рисунку 5.1.



```

Прямокутник зі сторонами a=3 і b=4
Об'єм V=0
Площа S=12
Паралелепіпед зі сторонами a=3, b=4, c=2
Об'єм V=24
Площа S=52

```

Рисунок 5.1 – Результати виконання програми

## Приклад 2

ООП реалізація з різною реалізацією внутрішньої архітектури та взаємодії класів

### Програмна реалізація

```
namespace oop_demo
{
    //Створюємо клас Point (точка на площині)
    class Point
    {
        //поля класу - координати точки на площині
        public int x, y;
        //конструктор класу
        public Point(int xx, int yy)// в конструктор
        передаються параметри для ініціалізації
        {
            //ініціалізація полів
            x = xx;
            y = yy;
        }
        //описуємо віртуальний метод - getLen() - отримати довжину
        public virtual double getLen()
        {
            return 0;// для точки поняття довжина не існує або
            рівна 0
        }
    }

    // Перший варіант реалізації класу MLine (відрізок)
    // в даному варіанті використано тип відносин між класами -
    композиція
    // об'єкти класу Point є невід'ємними частинами об'єкта MLine
    і визначають його початок та кінець
    class MLine
    {
        Point A, B;// полями класу є об'єкти типу Point
        public MLine(int x1, int y1, int x2, int y2)
        {
            //В конструкторі створюється два об'єкти Point
            //Викликаються конструктори класу Point
            A = new Point(x1, y1);
            B = new Point(x2, y2);
        }
        // Перевизначення віртуального методу - знаходження довжини
        відрізка
        public double getLen()
        {
            return Math.Sqrt(Math.Pow((A.x - B.x), 2) +
            Math.Pow((A.y - B.y), 2));
        }
    }
}
```

```

    }
}

// Другий варіант реалізації класу MLine1 (відрізок)
// в даному варіанті використано наслідування: Відрізок
наслідується від точки
class MLine1 : Point
{
    int x1, y1;
    public MLine1(int fx, int fy, int fx1, int fy1)
        : base(fx, fy) // виклик конструктора базового
класу
    {
        x1 = fx1;
        y1 = fy1;
    }
    //Перевизначення методу - знаходження довжини відрізка
    public override double getLen()
    {
        return Math.Sqrt(Math.Pow((x - x1), 2) +
Math.Pow((y - y1), 2));
    }
}
//тестування класів
class Program
{
    static void Main(string[] args)
    {
        MLine l1 = new MLine(0, 4, 3, 0); //перший варіант
реалізації
        Console.WriteLine("len l1={0}", l1.getLen());
        MLine1 l2 = new MLine1(0, 4, 3, 0); // другий
варіант реалізації
        Console.WriteLine("len l2={0}", l2.getLen());
        Console.ReadLine();
    }
}
}

```

Перший варіант реалізації здається більш природнім та логічним, оскільки точки кінців відрізка дійсно є невід’ємними елементами відрізка, точки кінців відрізка належать відрізку. Саме враховуючи математичний зміст понять «точка – відрізок» варіант з композицією є більш логічним. Хоча в окремих випадках другий варіант реалізації може мати свої переваги незважаючи на певну «нелогічність».

### **Завдання:**

Реалізувати ієрархію класів відповідно до варіанту, розробити демонстраційний модуль, що ілюструє створення та використання класів. Оформити звіт. Звіт повинне містити: титульний аркуш (назва та номер ЛР,

дані автора), мету роботи, індивідуальні завдання (завдання за варіантом), лістинг програмного коду, тестові запуски, відповіді на контрольні запитання, висновки (див. Додатки А, Б).

### Варіанти індивідуального завдання:

Варіанти	Завдання
1, 5, 9, 13	Точка ( <i>властивості</i> : координати; <i>методи</i> : відсутні) →Відрізок ( <i>властивості</i> : координати кінців; <i>методи</i> : довжина відрізка)→Трикутник ( <i>властивості</i> : координати вершин; <i>методи</i> : площа, периметр, порівняння з іншим об'єктом за площею)
2, 6, 10, 14	Точка ( <i>властивості</i> : координати; <i>методи</i> : відсутні) →Відрізок ( <i>властивості</i> : координати кінців; <i>методи</i> : довжина відрізка)→Прямокутник ( <i>властивості</i> : координати вершин; <i>методи</i> : площа, периметр, визначення чи є прямокутник квадратом)
3, 7, 11, 15	Точка ( <i>властивості</i> : координати; <i>методи</i> : відсутні) →Відрізок ( <i>властивості</i> : координати кінців; <i>методи</i> : довжина відрізка)→Круг ( <i>властивості</i> : координати центру, радіус; <i>методи</i> : площа, довжина кола, визначення чи є перетин з іншим об'єктом «кругом»)
4, 8, 12	Людина ( <i>властивості</i> : рік народження, стать, ім'я; <i>методи</i> : вік на даний момент) →Співробітник ( <i>властивості</i> : посада, дата прийняття, зар. платня; <i>методи</i> : стаж роботи в установі, розмір податку)→ Співробітник ІТ відділу ( <i>властивості</i> : спеціалізація, логін/пароль; <i>методи</i> : заміна логіна/пароля)

### Контрольні запитання:

1. Поясніть термін “ПАРАДИГМА”
2. Що таке клас в ООП?
3. Поясніть поняття: наслідування, поліморфізм, інкапсуляція.
4. Яким чином організовано класи з точки зору структури?

## *Додаток А*

### *Структура, зміст та вимоги до оформлення звіту з виконаної лабораторної роботи*

За результатами виконання лабораторної роботи здобувачем самостійно готується звіт, який повинен мати наступну структуру і зміст:

- титульний аркуш (Додаток Б);
- мета роботи, номер варіанту й завдання;
- самостійно одержані результати виконання завдань, перелік яких визначено порядком виконання лабораторної роботи;
- висновки (1/2 сторінки);
- додатки.

Оформлення звіту повинне відповідати наступним умовам:

- спосіб виконання – машинописний (комп'ютерний);
- параметри сторінки:
  - формат А4;
  - верхнє, нижнє, праве поле – 15 мм, ліве поле – 20 мм;
- текст:
  - шрифт Times New Roman ;
  - розмір шрифту – 14 пт;
  - міжабзацний інтервал – 0 пт;
  - міжрядковий інтервал – 1 пт;
  - абзацний відступ – 15 мм;
  - вирівнювання по ширині;
- лістинг (текст) програми:
  - шрифт Courier New;
  - розмір шрифту – 10 пт;
  - міжабзацний інтервал – 0 пт;
  - міжрядковий інтервал – 1 пт.

## *Додаток Б*

### *Титульний аркуш*

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

## Звіт

з лабораторної роботи № 1  
з дисципліни “Основи комп’ютерних технологій”  
на тему  
“ Системи числення ”

Виконав/ла  
здобувач/ка академічної групи

---

Прізвище й Ініціали

Перевірила  
старша викладачка  
Поліщук Л.І.



## **Додаток В**

### *Перелік тем рефератів*

1. Парадигми програмування
2. Основні поняття ООП
3. Розгляд прикладу побудови програми з використанням ООП
4. Аналіз задачі, декомпозиція
5. Моделювання, як фаза розробки програмного проекту
6. Моделі життєвого циклу
7. Поняття тестування
8. Тестування відкритого типу
9. Тестування закритого типу
10. Динамічні структури даних.
11. Лінійні зв'язані списки
12. Стек та черга
13. Дерево
14. Хеш-таблиці, пошук на основі хеш –таблиць
15. Класифікація підходів шифрування даних. Огляд вибраного алгоритму
16. Огляд типів структур даних сучасних мов програмування
17. Collection в мові C#. List, Set, Dictionary
18. Поняття баз даних, типи баз даних
19. Основні поняття реляційних БД
20. Модель «сутність –зв'язок», інфологічна модель
21. Нормалізація даних
22. Організація мережі, топології мереж
23. Поняття протоколів
24. WEB: сайт, структура сайту, елементи мови HTML, CSS
25. JavaScript та застосування мови при розробці web-додатків
26. Розробка моделі класів на прикладі конкретної задачі

#### **Вимоги до реферату:**

Об'єм 15-18 стр. Структура: Титульна сторінка, Зміст, Вступ, Основна частина, Висновки, Література, Додатки.

Мова виконання – українська.

Шрифт основного тексту: TimesNewRoman, 12 пт, 1,5 інт., абзац – 1,25.

Теоретична частина обов'язково має бути проілюстрована прикладами, фрагментами коду, конкретними прикладними задачами (відповідно до теми). Текст має бути перевірено на антиплагіат. Рівень оригінальності повинен бути не нижчим 60% (можливі сервіси для перевірки: Etxt – настільний додаток, Antiplagiat.ru – веб-сервіс). Перевірку своєї роботи здійснює кожен студент самостійно, в додатки додається звіт перевірки з відображенням показника оригінальності.

Реферат здається в двох примірниках: електронний документ (файл типу \*.doc, \*.docx) та друкований варіант (формат А4, підшитий в папку)

Реферат є обов'язковим видом роботи. Реферат здається в кінці курсу. Наявність зданих лабораторних робіт та захист реферату є умовою допуску студента до іспиту.