

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

А.Г. Кривохата, О.В. Кудін

ОСНОВИ ПРОГРАМУВАННЯ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Затверджено
вченою радою ЗНУ
Протокол № 6 від 24.12.2024

Запоріжжя
2024

УДК 004.4(076.5)
К821

Кривохата А. Г., Кудін О. В. Основи програмування : методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія». Запоріжжя : Запорізький національний університет, 2024. 101 с.

У методичних рекомендаціях подано теоретичні відомості та зміст лабораторних занять із курсу «Основи програмування». Розглянуто базові алгоритмічні конструкції, типи даних, принципи повторного використання коду, принципи роботи з динамічною пам'яттю комп'ютера. Значну увагу приділено засобам розробки та налагоджування власних програм.

До кожної теми наведено теоретичний матеріал, надано детальні рекомендації з виконання лабораторних робіт, запитання для самоперевірки.

Методичні рекомендації призначені для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія».

Рецензент

О. С. Пищенична, кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук

Відповідальний за випуск

А. О. Лісняк, кандидат фізико-математичних наук, доцент, завідувач кафедри програмної інженерії

Зміст

Вступ.....	4
Загальні рекомендації до виконання лабораторних робіт.....	6
Лабораторне заняття № 1. Програмування лінійних алгоритмів.....	7
Лабораторне заняття № 2. Програмування розгалужених алгоритмів.....	17
Лабораторне заняття № 3. Програмування циклічних алгоритмів.....	35
Лабораторне заняття № 4 Обробка масивів.....	48
Лабораторне заняття №5. Показчики. Створення динамічного масиву.....	60
Лабораторне заняття № 6. Символьний тип даних. Рядки	69
Лабораторне заняття № 7. Процедурне програмування.....	81
Лабораторне заняття № 8. Робота з файлами.....	92
Рекомендована література	99
Використана література	100

Вступ

Актуальність дисциплін, присвячених вивченню основ програмування обумовлена об'єктивною необхідністю успішного засвоєння майбутніми ІТ-фахівцями принципів опрацювання цифрової інформації, формування навичок алгоритмізації та програмування мовами високого рівня, вміння використовувати основні знання основ програмування при розробці інформаційних систем, що зараз охоплюють майже всі сфери життєдіяльності.

У навчально-методичному виданні подано короткий теоретичний матеріал та приклади створення програмних проєктів мовою програмування C. Розглянуто ключові аспекти, необхідні для формування фундаментальних знань та навичок у галузі розробки програмного забезпечення, зокрема: програмування лінійних, розгалужених і циклічних алгоритмів, робота зі статичними та динамічними масивами, обробка рядкових даних, основи процедурного програмування та взаємодія з файлами. Засвоєння цих тем є надзвичайно важливим, оскільки вони становлять основу для подальшого вивчення складніших концепцій програмування, таких як об'єктно-орієнтоване програмування, оптимізація алгоритмів і розробка багаторівневих програмних систем.

Метою вивчення навчальної дисципліни «Основи програмування» є: оволодіння теоретичними знаннями та набуття навичок з програмування та алгоритмізації, які дають можливість вирішувати задачі реалізації різних видів програмного забезпечення. Опанування даного курсу передуює вивченню курсів «Об'єктно-орієнтоване програмування», «Алгоритми і структури даних», «Організація комп'ютерних мереж», «Функціональне програмування».

Основними **завданнями** вивчення дисципліни «Основи програмування» є: засвоєння основних понять програмування, базових структур даних; ознайомлення з основними парадигмами програмування; опанування основних алгоритмів розв'язку типових задач та їх реалізація за допомогою конкретних мов програмування; набуття навичок розробки власних програм на мовах програмування для вирішення прикладних задач.

Згідно з вимогами освітньої-професійної програми здобувачі освіти мають досягти наступних компетентностей та програмних результатів навчання:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;
- здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
- здатність до алгоритмічного та логічного мислення;
- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;

- уміння вибрати та використовувати відповідну задачі методологію створення програмного забезпечення;
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- вміти застосовувати методи компонентної розробки програмного забезпечення.

Структура та вміст методичних рекомендацій відповідають змісту силабуса дисципліни «Основи програмування».

Навчально-методичне видання містить вісім лабораторних робіт, контрольні запитання, посилання на інформаційні джерела, які можна використовувати при розробці програм, переліки рекомендованої та використаної літератури.

Загальні рекомендації до виконання лабораторних робіт

Лабораторні заняття з дисципліни «Основи програмування» призначені для того, щоб студент під керівництвом викладача, особисто проводив імітаційні експерименти, з метою практичного підтвердження окремих теоретичних положень навчальної дисципліни; набував практичних навичок програмування.

Відповідно до власного варіанта (визначається за списком академгрупи) студент має виконати завдання лабораторних робіт. Наприкінці кожної лабораторної роботи наведено посилання на інформаційні джерела, а також контрольні запитання, відповіді на які потрібно додати у звіт з виконання лабораторної роботи.

Результати виконання роботи необхідно продемонструвати на лабораторному занятті під час захисту. Захист лабораторної роботи відбувається таким чином: студент пояснює та демонструє етапи реалізації завдання та отримані результати, відповідає на запитання. Захист роботи – це презентація своїх результатів за допомогою комп'ютера, а також практичне виконання деяких елементів роботи у режимі реального часу.

При виконанні кожної роботи необхідно ознайомитись із загальними теоретичними відомостями та матеріалами відповідної лекції (деякі моменти потрібно опрацювати самостійно, користуючись переліком рекомендованої літератури); виконати практичні завдання. За результатами роботи необхідно оформити звіт, який повинен містити:

- титульну сторінку;
- номер варіанта;
- умови завдань, лістинги програм та результати роботи;
- відповіді на контрольні запитання.

Загальний шаблон цього документу розташовано на сторінці курсу «Основи програмування» в системі Moodle.

Готовий звіт завантажується до СЕЗН Moodle у відведені викладачем терміни. Назва файлу, що завантажується, має вигляд: Прізвище_Ім'я_ІrN (де N – номер лабораторної роботи) та формат текстового документу (приймаються розширення doc, docx, pdf, odt). Якщо розмір файлів перевищує 5Мб, то відповіді розміщуються на зовнішніх ресурсах, а до СЕЗН додається відповідне посилання.

Лабораторне заняття № 1. Програмування лінійних алгоритмів

Мета: ознайомитись із загальними правилами створення програм мовою С та набутти практичних навичок програмування лінійних алгоритмів.

План

1. Блок-схеми програм.
2. Структура С-програми. Основи синтаксису мови програмування С.
3. Приклад задачі з лінійною структурою.
4. Основні заголовні файли С.
5. Завдання до лабораторної роботи.
6. Контрольні запитання.
7. Корисні посилання.

1.1 Блок-схеми програм

Одним зі способів зображення алгоритмів є графічне зображення алгоритма, тобто блок-схема. При такому способі кожна дія записується всередині блоків. Блоки з'єднуються лініями, які вказують на послідовність дій. Лінії можуть закінчуватись стрілками, але зазвичай стрілки не ставлять, якщо лінії відображають природню послідовність дій. В іншому разі стрілки обов'язкові. Природня послідовність: зверху вниз та зліва направо. Розглянемо декілька прикладів.

Приклад 1.1 Скласти блок-схему до задачі: за заданою стороною рівностороннього трикутника обчислити його периметр.

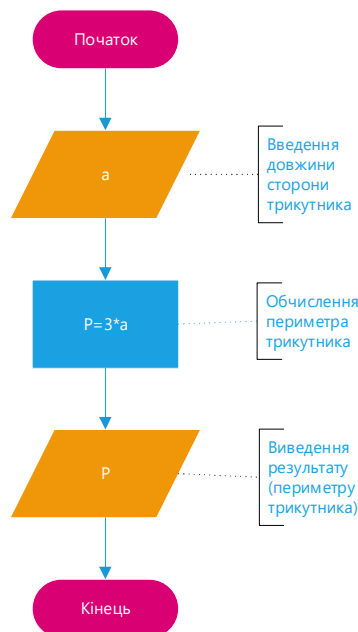


Рисунок 1.1 – Блок-схема алгоритму обчислення периметру трикутника

Приклад 1.2 Скласти блок-схему до задачі: перевірити ціле число щодо подільності на 7.

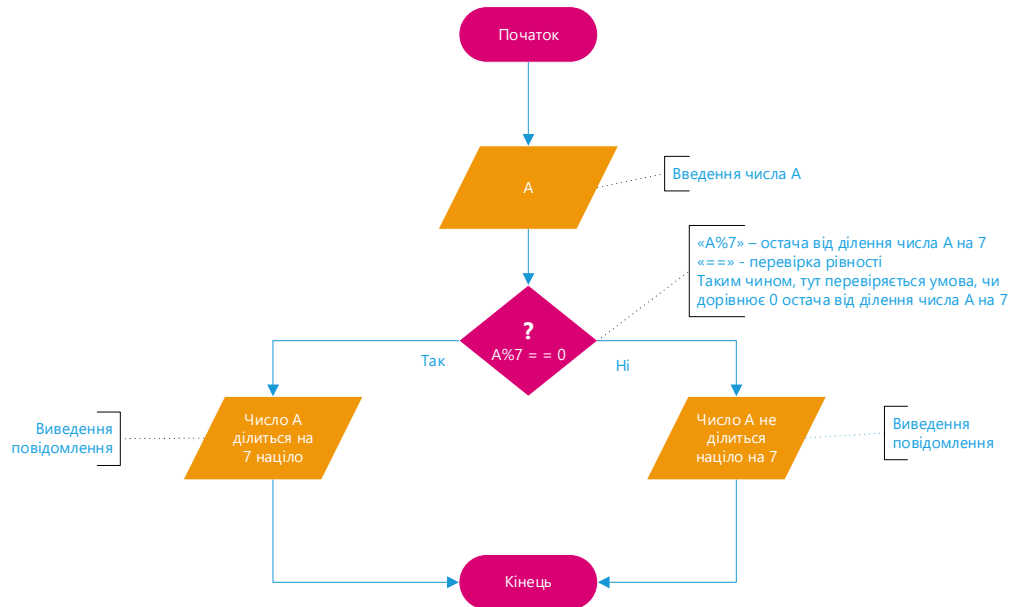


Рисунок 1.2 – Блок-схема алгоритма з розгалуженням

1.2 Структура програми. Основи синтаксису мови С

На рисунку 1.3 зображено структуру вікна IDE Visual Studio.

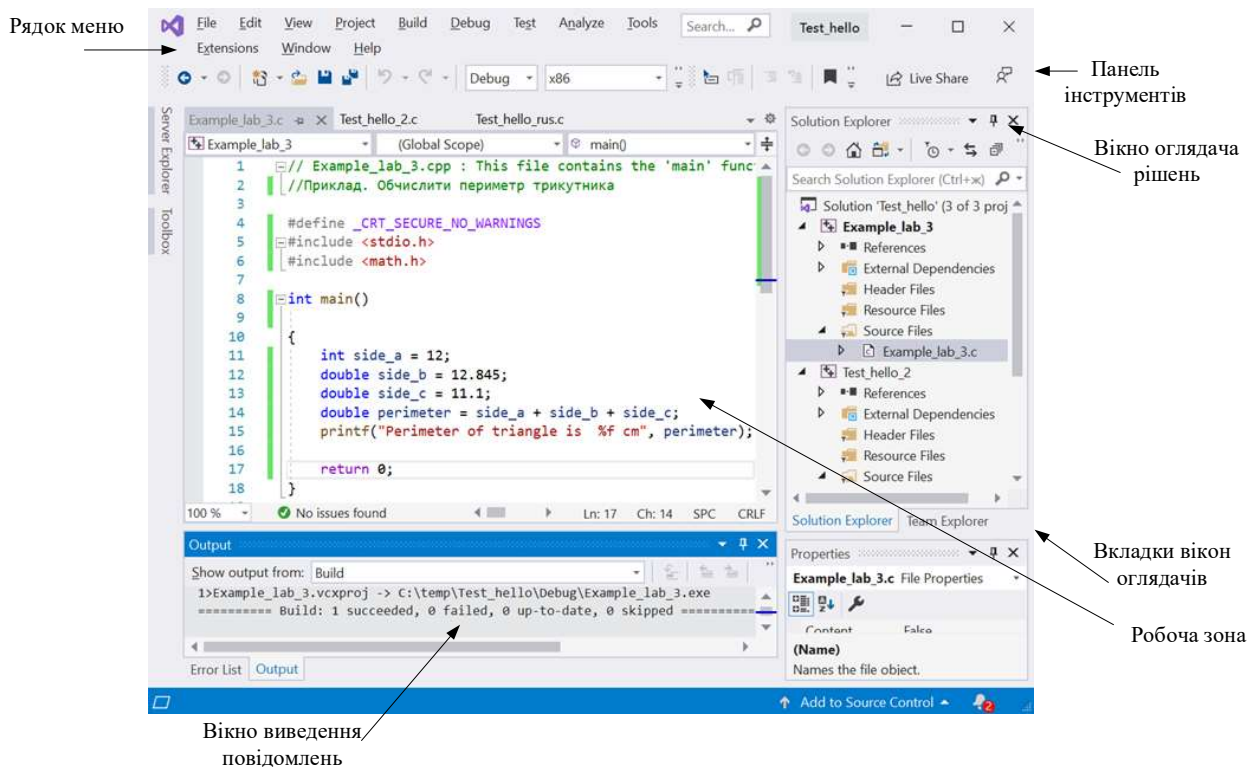


Рисунок 1.3 – Структура вікна IDE Visual Studio

Для запуску програми попередньо необхідно виконати її компіляцію та збірку відповідним функціоналом головного меню. Також у Visual Studio існують засоби налагодження програм, а саме, виконання до встановленої контрольної точки, покрокове виконання, перегляд значення змінних під час виконання програми.

Будь-яка програма, що створено мовою програмування C, складається мінімум з однієї функції **main()**, яка є головною. Тіло функції записується у фігурних дужках.

// – таким знаком відокремлюється **коментар**

/* у такій конструкції можна написати **багаторядковий коментар** */

```
#include <stdio.h> //підключення заголовного файлу
int main( ) /*проста програма*/
{
    cout << "Hello world\n"; //неформатоване виведення на екран
}
```

Рисунок 1.4 – Приклад оформлення коментарів

#include <stdio.h> – підключення іншого файлу. Цей рядок вказує компілятору, що потрібно задіяти інформацію, що міститься у файлі **stdio.h** (ця конструкція називається **директивою**).

1.3 Приклад задачі з лінійною структурою

Розглянемо приклад, в якому використовуватиметься введення даних з клавіатури. Для цього буде використано бібліотечну функцію **scanf_s()**, яка дозволяє вводити інформацію з клавіатури під час виконання програми (див. рис. 1.5).

```
3      /*Приклад: обчислення довжини кола*/
4
5      #include <stdio.h>
6
7      main()
8      {
9          int radius;
10         float lenght;
11         printf ("Enter value of radius: \n");
12         scanf_s("%d", &radius);
13         lenght = 3.1415 * 2 * radius;
14         printf("Радіус - %d\n, Довжина - %f\n", radius, lenght);
15     }
```

Рисунок 1.5 – Лістинг програми. Обчислення довжини кола

Наведемо декілька пояснень до цієї програми.

По-перше, оголошено змінні двох різних типів:

radius – типу ціле (**int**);

length – типу з плаваючою комою (**float**), що містить дробну частину.

По-друге використовується функція **scanf_s()** для введення з клавіатури значення радіуса кола. Перший аргумент функції **scanf_s()** - **%d** показує, що буде введено ціле десяткове число. Другий аргумент – ім'я змінної, якій буде привласнено введене значення. Символ **&** (амперсанд) перед іменем змінної **radius** необхідний для правильної роботи функції **scanf_s()**. Більш докладно використання цього символу обговорюватиметься пізніше.

В наступному рядку програми цілі числа **2** та **radius** множаться на число з плаваючою комою **3.1415** і результат привласнюється змінній **length** типу **float**. Мова С дає можливість використання у виразах змінних різних типів. Для виведення результатів використовується функція **printf()**, яка містить 4 аргументи. Специфікатор формату **%f** використовується для друку значення змінної **length** типу **float**.

В розглянутому прикладі довжина кола обчислюється тільки для цілочисельних значень радіуса. Для того, щоб програма могла обчислювати довжину кола будь-якого радіусу, необхідно оголосити змінну **radius** як **float** або **double**, а в функції **scanf_s()** використовувати специфікатор **%f**

1.4 Основні заголовні файли С

В таблиці 1.1 наведено перелік та вміст основних заголовних файлів мови програмування С.

Таблиця 1.1 – Основні заголовні файли С

Назва	Призначення
conio.h	Функції консольного введення виведення
ctype.h	Функції для перевірки літер
math.h	Математичні функції та макроси
stdio.h	Функції введення-виведення, типи, макроси
stdlib.h	Функції перетворення чисел, запиту пам'яті тощо
string.h	Функції для роботи з рядками символів
time.h	Типи і функції пов'язані з датою і числом

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №1

Відповідно до варіанта (визначається за списком академгрупи) виконати 3 завдання. Умови та варіанти наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1, 2, 3;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту:

1. Титульна сторінка.
2. Номер варіанта.
3. Умови завдань 1, 2, 3, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №1

Завдання 1. Запишіть математичний вираз алгоритмічною мовою, вкажіть послідовність виконання комп'ютером арифметичних дій

Примітка. Відповідь до цього завдання орієнтовно виглядатиме так:

$$U = \frac{e^{k+y} + \operatorname{tg}(x\sqrt{y})}{x-d^2}, U = (\exp(k+y) + \tan(x*\sqrt{y})) / (x - \operatorname{pow}(d, 2))$$

№ вар.	Вираз	№ вар.	Вираз
1	$G = \frac{e^{2y} + \sin f}{\ln(3,8y + f)}$	21	$W = \frac{4t^3 + \ln r}{e^{y+r} + 7,2 \sin r}$
2	$F = \ln d + \frac{3,5d^2 + 1}{\cos 2yr}$	22	$H = \frac{y^2 - 0,8y + \sqrt{y}}{23,1n^2 + \cos n}$
3	$U = \frac{\ln(k - y) + y^4}{e^y + 2,355k^2}$	23	$R = \frac{\sqrt{\sin^2 y^2 + 6,835}}{\ln(y + k) + 3y^2}$
4	$G = \frac{9,33w^2 + \sqrt{w}}{\ln(y + 3,5) + \sqrt{y}}$	24	$E = \frac{\ln(0,7y + 2q)}{\sqrt{3y^2 + 0,5y + 4}}$
5	$D = \frac{7,8a^2 + 3,52t}{\ln(a + 2y) + e^y}$	25	$K = \frac{2t^2 + 3t + 7,2}{\ln y + e^{2y}}$
6	$L = \frac{0,81 \cos i}{\ln y + 2y^3}$	26	$Q = \frac{\sqrt{k + 2,6p \sin k}}{x - d^2}$
7	$N = \frac{m^2 + 2,8m + 0,355}{\cos 2y + 3,6}$	27	$S = \frac{4,351y^3 + 2t \ln t}{\sqrt{\cos 2y + 4,351}}$
8	$T = \frac{2,37 \sin(t + 1)}{\sqrt{4y^2 - 0,1y + 5}}$	28	$R = \frac{\sin^2 y + 0,3d}{e^y + \ln d}$
9	$V = \frac{(y + 2w)^3}{\ln(y + 0,75)}$	29	$U = \frac{\ln(2k + 4,3)}{e^{k+y} + \sqrt{y}}$
10	$Q = \frac{2t + y \cos t}{\sqrt{y + 4,831}}$	30	$L = \cos^2 c + \frac{3t^2 + 4}{\sqrt{c + t}}$

№ вар.	Вираз	№ вар.	Вираз
11	$D = y^2 + \frac{0,5n + 4,8}{\sin y}$	31	$T = \frac{\sin 2u}{\ln(2y + u)}$
12	$R = \frac{\sin(2t + 1)^2 + 0,3}{\ln(t + y)}$	32	$H = \frac{\sin(p + 0,4)^2}{y^2 + 7,325p}$
13	$A = \frac{\sin(2y + h) + h^2}{e^h + y}$	33	$W = \frac{0,004v + e^{2y}}{e^{\frac{y}{2}}}$
14	$P = \frac{e^{y+2,5} + 7,1h^3}{\ln \sqrt{y + 0,04h}}$	34	$T = \frac{0,355h^2 - 4,355}{e^{y+h} + \sqrt{2,7y}}$
15	$F = \frac{2 \sin(0,354 + 1)}{\ln(y + 2j)}$	35	$N = \frac{3y^2 + \sqrt{y + 1}}{\ln(y + p) + e^p}$
16	$R = \frac{3t^2 + 3t^5 + 4,9}{2 \sin(0,354 + 1)}$	36	$S = \frac{\sqrt{\cos(4y^2) + 7,151}}{3t^2 + 3t^5 + 4,9}$
17	$K = \frac{\ln(p^2 + y^3) + e^p}{\ln(t + y)}$	37	$N = \frac{\sin(2t + 1)^2 + 0,3}{\ln(yp) + e^p}$
18	$G = \frac{n(y + 3,5) + \sqrt{y}}{\sqrt{c + t}}$	38	$P = \frac{3y^2 + \sqrt{y^3 + 1}}{\cos 2y + 3,6}$
19	$D = \frac{9,8a^2 + 5,52 \cos t^5}{\ln(0,7y + 2q)}$	39	$Q = \frac{n\sqrt{y^3 + 1,09g}}{23,1n^2 + \cos n}$
20	$L = \frac{1,51 \cos x^5 + 2x^3}{\cos 2y + 3,6}$	40	$U = \frac{e^{k+y} + \tan(x\sqrt{y})}{x - d^2}$

Завдання 2. Розробити блок-схему та програму (у середовищі Visual Studio або ін. IDE) для свого варіанта (див. нижче). Обов'язковим є використання форматованого введення та виведення даних, код має бути оформлено відповідно «C++ Programming Style Guidelines». Всі дані вводити з клавіатури та передбачити консольне меню для введення даних.

Примітка! Назви Solution для всіх лабораторних робіт мають складатися з прізвища латиницею та номера роботи, назви проєктів мають відповідати номерам завдань, наприклад Task1.

№ вар.	Завдання
1	За заданим радіусом кола обчислити та вивести на екран довжину кола та площу круга.
2	За заданим радіусом кулі обчислити та вивести на екран її об'єм та площу поверхні.
3	За заданими сторонами прямокутника обчислити та вивести на екран його площу та периметр.
4	За заданими сторонами паралелепіпеда обчислити та вивести на екран його об'єм та площу поверхні.
5	За заданими значеннями основ та висоти рівнобічної трапеції обчислити та вивести на екран її площу та периметр.
6	За заданими радіусом основи та висотою прямого кругового циліндра обчислити та вивести на екран його об'єм та площу поверхні.
7	За заданими катетами прямокутного трикутника обчислити та вивести на екран його площу та периметр.

№ вар.	Завдання
8	У прямому круговому конусі відомі радіус основи та висота. Обчислити та вивести на екран його об'єм та площу поверхні.
9	Відома сторона довільного правильного n -кутника. Обчислити та вивести на екран його площу та периметр.
10	Задані три сторони довільного трикутника. Обчислити та вивести на екран його площу та периметр.
11	Відома сторона правильного трикутника. Обчислити та вивести на екран його площу, радіуси вписаного та описаного кіл.
12	Відомі сторона та висота правильної чотирикутної піраміди. Обчислити та вивести на екран її об'єм та площу поверхні.
13	Обчислити і вивести на екран катет і площу прямокутного трикутника. Довжину іншого катета і гіпотенузи ввести з клавіатури.
14	Обчислити та вивести на екран довжину відрізка, кінці якого задано координатами у декартовій системі координат. Координати ввести з клавіатури.
15	Розробити програму, яка обчислює об'єм циліндра та конуса, які мають однакові висоту H та радіус основи R . Дані користувач вводить з клавіатури.
16	Дано довжину ребра куба. Знайти об'єм куба та площу його бічної поверхні. Необхідні дані ввести з клавіатури.
17	Дано два дійсних числа додатніх числа. Знайти середнє арифметичне та середнє геометричне цих чисел. Числа ввести з клавіатури
18	Задано катети прямокутного трикутника. Знайти його гіпотенузу та площу. Необхідні дані ввести з клавіатури.
19	Визначити час падіння каменя на поверхню землі з висоти h . Необхідні дані ввести з клавіатури.
20	Обчислити період коливання маятника довжини l . Необхідні дані ввести з клавіатури.
21	Обчислити силу притягання F між тілами маси m_1 та m_2 , які знаходяться на відстані r одне від одного. Необхідні дані ввести з клавіатури.
22	Задано відстань L в сантиметрах. Знайти кількість повних метрів в ньому, використовуючи операцію ділення націло. Необхідні дані ввести з коавіатури.
23	Обчислити довжину кола та площу круга заданого радіуса R . Необхідні дані ввести з клавіатури.
24	Обчислити площу прямокутного трикутника для заданих значень двох його катетів a і b . Необхідні дані ввести з клавіатури.
25	Обчислити висоту правильного трикутника h , якщо відомо сторону a трикутника. Необхідні дані ввести з клавіатури.
26	Знайти значення функції $y = 3x^6 - 6x^2 - 7$ при заданому (ввести з клавіатури) значенні x .
27	З початку доби минуло N секунд (N - ціле). Використовуючи операцію ділення націло знайти кількість повних хвилин, які минули з початку доби.
28	Задано (ввести з клавіатури) розмір файлу у байтах. Використовуючи операцію ділення націло знайти кількість повних кілобайтів, які займає даний файл.
29	Обчислити площу трикутника за трьома сторонами – a , b , c . Довжни сторін ввести з клавіатури.
30	Задано два числа. Знайти середнє арифметичне кубів цих чисел та середнє геометричне модулів цих чисел.
31	Обчислити діаметр кола за заданою (користувач вводить з клавіатури) довжиною кола. Константу π «втягти» з бібліотеки <code>math.h</code>
32	Обчислити висоту правильного трикутника h за заданою стороною a правильного

№ вар.	Завдання
	трикутника.
33	Дано два невід'ємних числа a і b . Знайдіть їх середнє геометричне.
34	Дано три числа a, b, c . Знайдіть середнє арифметичне квадратів цих чисел.
35	Відома сторона довільного правильного n -кутника. Обчислити та вивести на екран його площу та периметр.
36	За заданим радіусом кулі обчислити та вивести на екран її об'єм та площу поверхні.
37	Розробити програму, яка обчислює об'єм циліндра та конуса, які мають однакові висоту H та радіус основи R . Дані користувач вводить з клавіатури.
38	За заданими значеннями основ та висоти трапеції обчислити та вивести на екран її площу та периметр.
39	За заданим радіусом кола обчислити та вивести на екран довжину кола та площу круга.
40	За заданим радіусом кулі обчислити та вивести на екран її об'єм та площу поверхні.

Завдання 3. Складіть структурну схему алгоритма та програму обчислення значення функції $F(x)$, один з параметрів ввести з клавіатури, а інший – задати як константу. Всі обчислені значення вивести на екран.

№ вар.	Функція $y=F(x)$	Значення параметрів	№ вар.	Функція $y=F(x)$	Значення параметрів
1	$y = e^{2x} + 9.7^c$ $c = a^2 + \sqrt{b * x}$ $a = \lg x$	$x = 3.5$ $b = 0.4$	21	$y = \cos^2 b + b \cos^4 a^2$ $a = \sqrt[3]{ b + c }$ $b = x + c^2$	$x = 8.52$ $c = 9$
2	$y = xa^3 + b^k$ $a = \ln x $ $b = e^{2x} + ax$	$x = 1.3$ $k = 4$	22	$y = \cos^3(a + b^3)$ $a = tx + \sqrt{ b }$ $b = \log^2 x $	$x = 0.9$ $t = 2$
3	$y = a^3/b^2$ $a = e^{\sqrt{ x }}$ $b = (\sin p^2 + x^3)$	$x = 2.1$ $p = 1$	23	$y = a^3/\cos c$ $c = \log a + b^2$ $a = \sqrt{ x } + e^{\sqrt{b}}$	$x = -8$ $b = 9.5$
4	$y = \sqrt[5]{ a - bx }$ $a = \log x$ $b = \sqrt{x^2 + t^2}$	$x = 2.7$ $t = -6$	24	$y = xp^2 + t^5$ $p = x^2 - \sqrt{ x }$ $t = x^2 + a^2$	$x = 4$ $a = 3.7$
5	$y = \operatorname{tg}^3 x^2 + kt$ $x = pt^2 + \sqrt{k}$ $k = \sqrt{pt}$	$t = 4.1$ $p = 3$	25	$y = \sin^3 ax + \sqrt{b} \cos x^2$ $a = \ln x $ $b = x^4 + \lg p^3$	$x = 1.4$ $p = 1.6$
6	$y = \sin^2(a + \operatorname{tg}^3 b)$ $a = \sqrt{ x }$ $b = x^4 + m^2$	$m = 2$ $x = 1.1$	26	$y = \ln a / \lg b^3$ $a = \sin(x^2 + b^2)$ $x = n^b + b^2$	$n = 8.1$ $b = 1.6$
7	$y = \lg^3 a + \cos^5 x$ $a = c^4 + k^3$ $c = \sqrt{ x }$	$k = 8.2$ $x = 5$	27	$y = \lg^4 x + t $ $x = p^2 + t$ $t = \sin m^3$	$m = 5.7$ $p = 4$
8	$y = 7e^{\sqrt{ x }} + \cos^4 x$ $x = a + (b + c)^3$ $a = \sin b$	$b = 2.2$ $c = 3.7$	28	$y = \ln(a + b) + \frac{a^2}{a + t}$ $a = \lg tx + b^2 $ $t = xb^2 + \sqrt{x}$	$b = 0.3$ $x = 5.2$
9	$y = \sqrt[3]{x^2 + c^2}$	$k = 1$ $m = 1.8$	29	$y = \cos x^5 - b \sin^2 x$	$t = 6.2$ $b = 1.8$

№ вар.	Функція $y=F(x)$	Значення параметрів	№ вар.	Функція $y=F(x)$	Значення параметрів
	$x = e^{mk}$ $c = \cos^2 m + k^2$			$x = a^3 + \sqrt{t+b}$ $a = t^2 b$	
10	$y = e^x + 5.8^c$ $c = a^2 + \sqrt{b}$ $a = b^3 + \ln b $	$x = 2.8$ $b = 1.3$	30	$y = x^3/t^2$ $x = e^{\sqrt{p+a}}$ $t = p^3 + a^3$	$a = 2$ $p = 2.6$
11	$y = c^2 + \sqrt{ a }$ $c = \lg b $ $a = (b+x)^3$	$b = 7$ $x = 2$	31	$y = x^2 + \sqrt[3]{ x }$ $x = \cos^2 b + \sin^2 a$ $a = \sqrt{b+t^2}$	$b = 8.1$ $t = 2$
12	$y = \operatorname{arctg}^2 x $ $x = t^3 + b^2$ $t = b^3 + e^{\sqrt{q}}$	$q = 2$ $b = 1.8$	32	$y = \operatorname{tg} 4x + \sin x^2$ $x = \ln ct + a^2$ $c = t^2 + \sqrt{a}$	$t = -3$ $a = 76$
13	$y = at^2 + \sin^2 w$ $t = \cos^2 a + w^2$ $w = x^2 \sqrt{ a+x }$	$x = 1.9$ $a = -0.9$	33	$y = \cos^4(a^3 + b)$ $a = b + t^2 e^t$ $t = k^2 + \sqrt{b}$	$b = 2.19$ $k = 1.7$
14	$y = \sin^4(a^2 + b^2)$ $a = \sqrt{b+t}$ $t = b^2 + k^3$	$b = 6$ $k = 3.4$	34	$y = \lg^3 p + x$ $p = x(b^3 + a)$ $x = e^{2a} + \sqrt{b}$	$a = 6$ $b = 4.3$
15	$y = \cos^3 x + a $ $x = e^b$ $b = a + \sqrt{a+p^2}$	$a = 5.5$ $p = 4$	35	$y = \ln^5 x+a^2 $ $x = t^2 \sqrt{ a+b }$ $a = t + b^3$	$t = 3$ $b = 4.2$
16	$y = e^{2x} + 9.7^c$ $c = a^2 + \sqrt{b * x}$ $a = \lg x$	$x = 3.5$ $b = 0.4$	36	$y = \cos^2 b + b \cos^4 a^2$ $a = \sqrt[3]{ b+c }$ $b = x + c^2$	$x = 8.52$ $c = 9$
17	$y = a^3/b^2$ $a = e^{\sqrt{ x }}$ $b = (\sin p^2 + x^3)$	$x = 2.1$ $p = 1$	37	$y = a^3/\cos c$ $c = \log a + b^2$ $a = \sqrt{ x } + e^{\sqrt{b}}$	$x = -8$ $b = 9.5$
18	$y = \operatorname{tg}^3 x^2 + kt$ $x = pt^2 + \sqrt{k}$ $k = \sqrt{pt}$	$t = 4.1$ $p = 3$	38	$y = \sin^3 ax + \sqrt{b} \cos x^2$ $a = \ln x $ $b = x^4 + \lg p^3$	$x = 1.4$ $p = 1.6$
19	$y = \lg^3 a + \cos^5 x$ $a = c^4 + k^3$ $c = \sqrt{ x }$	$k = 8.2$ $x = 5$	39	$y = \lg^4 x+t $ $x = p^2 + t$ $t = \sin m^3$	$m = 5.7$ $p = 4$
20	$y = 7e^{\sqrt{ x }} + \cos^4 x$ $x = a + (b+c)^3$ $a = \sin b$	$b = 2.2$ $c = 3.7$	40	$y = \ln(a+b) + \frac{a^2}{a+t}$ $a = \lg tx + b^2 $ $t = xb^2 + \sqrt{x}$	$b = 0.3$ $x = 5.2$

Контрольні запитання

1. Що означає тип даних?
2. Які стандартні типи даних є у мові C?
3. Що таке переповнення комірки оперативної пам'яті?
4. У чому полягає відмінність операцій / та % ?

5. Які аргументи використовуються у функціях `scanf()` та `printf()`?
6. В природній мові є слова та вирази, які називаються жаргонами. Наведіть приклади жаргонів у мовах програмування.

Корисні посилання

1. Зручні сервіси для створення блок-схем. URL: <https://robotdreams.cc/uk/blog/198-udobnyh-servisov-dlya-sozdaniya-blok-shem>
2. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
3. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html>
4. Const, constexpr і символічні константи. URL: <https://acode.com.ua/urok-40-const-constexpr-i-symvolni-konstanty/>
5. Форматоване введення-виведення C. URL: <http://cpp.dp.ua/formatovane-vvedennya-vyvedennya/>

Лабораторне заняття № 2. Програмування розгалужених алгоритмів

Мета: ознайомитись з видами розгалужень в мові C та набути практичних навичок програмування розгалужених алгоритмів.

План

1. Логічні операції в мові програмування C.
2. Оператор розгалуження **if**.
3. Оператор вибору варіантів **switch**.
4. Завдання до лабораторної роботи.
5. Контрольні запитання.
6. Корисні посилання.

2.1 Логічні операції в мові програмування C

В мові програмування C наявні три логічні операції:

1. Логічна операція «І», позначається як **&&**;
2. Логічна операція «АБО», позначається як **|**;
3. Логічна операція «НЕ» або логічне заперечення, позначається як **!**.

У таблиці 2.1 коротко охарактеризовано логічні операції в мові програмування C, що використовуються для побудови логічних умов.

Таблиця 2.1 – Опис логічних операцій

Операція	Позначення	Приклад	Короткий опис
I	&&	$a == 3 \ \&\& \ b > 4$ (<i>a</i> дорівнює 3 І <i>b</i> >4)	Складена умова істинна, якщо істинні обидві простих умови
АБО		$a == 3 \ \ b > 4$ (<i>a</i> дорівнює 3 АБО <i>b</i> >4)	Складена умова істинна, якщо істинна, хоча б одна з простих умов
НЕ	!	$a != 3$ (<i>a</i> НЕ дорівнює 3)	Умова істинна, якщо <i>a</i> не дорівнює 3

Приклад 2.1 Записати таблицю істинності для логічного виразу **$a \ \&\& \ (!a \ || \ b)$**

Для створення такої таблиці необхідно розглянути всі варіанти значень логічних змінних **a** та **b** (можуть приймати значення лише **true** або **false**). Отже створюємо таблицю, куди вносимо всі можливі комбінації вхідних змінних (перші два стовпці). Далі додаємо стовпці, що містять чергову логічну операцію з урахуванням пріоритету. У нашому виразі спочатку виконується операція заперечення у дужках (третій стовпець), потім логічне «або» у дужках (четвертий стовпець) і наприкінці логічне «і». По-черзі для кожного стовпця обчислюємо значення виразу, таким чином останній стовпець містить значення всього виразу для визначених значень змінних.

a	b	!a	!a b	a && (!a b)
false	false	true	true	false
false	true	true	true	false
true	false	false	false	false
true	true	false	true	true

2.2 Оператор розгалуження **if**

Приклад 2.2 Розробити програму для розв'язання квадратного рівняння із заданими користувачем коефіцієнтами

```

9  int main()
10 {
11     double koef_a, koef_b, koef_c;
12     double Diskriminant;
13     double root_1, root_2;
14     printf("Enter coefficients of equation\n");
15     scanf_s("%f %f %f", &koef_a, &koef_b, &koef_c);
16     Diskriminant = koef_b*koef_b - 4*koef_a*koef_c;
17     if (Diskriminant < 0)
18         printf("No real roots");
19     else {
20         root_1 = ((-1)*koef_b - sqrt(Diskriminant))/(2*koef_a);
21         root_2 = ((-1)*koef_b + sqrt(Diskriminant))/(2*koef_a);
22         printf("Root1 = %f\nRoot2 = %f\n", root_1, root_2);
23     }
24     return 0;
25 }
```

Рисунок 2.1 – Програма обчислення коренів квадратного рівняння

Наведемо декілька коментарів до цієї програми.

Рядки 11-13 – оголошуються змінні (коефіцієнти квадратного рівняння, дискримінант, корені), визначається тип (в даному випадку **double**). Рядок 15 – користувач вводить три числа (коефіцієнти квадратного рівняння), значення яких привласнюються змінним **koef_a**, **koef_b**, **koef_c** відповідно. Далі можна обчислювати дискримінант за всім відомою формулою. Рядок 17 – перевіряється умова, чи є отримане значення дискримінанту меншим за нуль. Якщо так, то виводиться відповідне повідомлення (рядок 18). Якщо умова (рядок 17) не є істинною, то виконуватиметься набір операторів після конструкції **else** (рядки 20-22). Зверніть увагу на виведення результатів на екран (рядок 22).

2.3 Оператор вибору варіантів **switch**

Приклад 2.3 Розробити програму для обчислення значення виразу:

$$s = \begin{cases} |f(x)| + \ln(y), & |xy| > 10 \\ e^{f(x)+y}, & 5 < |xy| \leq 10 \\ \sin(x) + \tan(y), & |xy| = 5 \end{cases}$$

При виконанні завдання запропонувати вибір виду функції $f(x)$: $sh(x)$, x^2 або e^x .

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <Windows.h>
4
5  int main() {
6      SetConsoleCP(1251);
7      SetConsoleOutputCP(1251);
8      float x, y, f, a, s;
9      int key;
10     //Введення даних
11     printf("Введіть значення x:");
12     scanf_s("%f", &x);
13     printf("Введіть значення y:");
14     scanf_s("%f", &y);
15     printf("Оберіть вид функції f(x): 1 - sh(x), 2 - x^2, 3 - exp(x):");
16     scanf_s("%i", &key);
17     switch (key) {
18         case 1: f = sinh(x); break;
19         case 2: f = pow(x, 2); break;
20         case 3: f = exp(x); break;
21         default: printf("Вибір не вірний"); return 1;
22     }
23     a = fabs(x * y);
24     if (a < 5) {
25         printf("Результат відсутній");
26         return 1;
27     }
28     else
29         if (a > 10) s = fabs(f) + log(y);
30         else
31             if (a <= 10 && a > 5) s = exp(f + y);
32             else s = sin(x) + tan(y);
33     printf("Значення функції: %7.3f ", s);
34     return 0;
35 }

```

Рисунок 2.2 – Програма обчислення значення виразу

Наведемо декілька коментарів до цієї програми. По-перше, для коректного відображення та зчитування кирилиці в консолі використано функції **SetConsoleCP(1251)** та **SetConsoleOutputCP(1251)** із використанням відповідного кодування. Для реалізації вибору функцій з умови завдання застосовано оператор **switch**, ключ до якого вводить користувач з клавіатури (рядок 16). В залежності від значення змінної **key** відбувається присвоєння змінній **f** значення відповідної функції (рядки 18-20). Також передбачено обробку ситуації, коли користувач вводить некореткне значення ключа (тобто такого значення, яке відсутнє серед параметрів **case** – рядок 21). Далі переходимо до безпосереднього обчислення значення виразу і в залежності від значення добутку $|xy|$ обчислюється значення функції **s**.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №2

Відповідно до варіанта (визначається за списком академгрупи) виконати 4 завдання. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1a, 1b, 2, 3, 4;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта.
3. Умова завдання 1a та розв'язання.
4. Умова завдання 1b та розв'язання (таблиця істинності).
5. Умови завдань 2-4, тексти та результати роботи програм.
6. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №2

Завдання 1a. Обчислити значення логічного виразу та вказати порядок виконання дій.

№	Вираз	№	Вираз
1	$(2*2==4) \ \&\& \ (3>5)$	7	$(A<=A+1) \ \ (12-3>0)$
2	$(2*2==4) \ \ (4<2)$	8	$((2<=2) \ \&\& \ (3>=3)) \ \ (15>25)$
3	$!(1<10) \ \ (4>25)$	9	$(6<4-2) \ \&\& \ !(3+5>0)$
4	$(1==1) \ \&\& \ (0==0)$	10	$(i<=i-1) \ \ !(7-10>0)$
5	$(i>=i-1) \ \ (3-5>0)$	11	$(4*3==6) \ \&\& \ !(7-5<=0)$
6	$!((6<4) \ \&\& \ (4>8))$	12	$(2<=0) \ \&\& \ (i>=i+1) \ \ (15>25-5)$
13	$(2*2==4) \ \&\& \ !(3+5>0)$	14	$!(1<10) \ \ (12-3>0)$
15	$(1==1) \ \&\& \ !(3+5>0)$	16	$(4*3==6) \ \&\& \ (4>8)$
17	$((2<=2) \ \&\& \ (3>=3)) \ \ (15>25)$	23	$!(1<10) \ \ (4>25)$
18	$(i<=i-1) \ \ !(7-10>0)$	24	$(2*2==4) \ \&\& \ !(3+5>0)$
19	$(4*3==6) \ \&\& \ !(7-5<=0)$	25	$(1==1) \ \&\& \ !(3+5>0)$
20	$(6<4-2) \ \&\& \ !(3+5>0)$	26	$!(1<10) \ \ (4>25)$
21	$(A<=A+1) \ \ (12-3>0)$	27	$(i>=i-1) \ \ (3-5>0)$
22	$(2<=0) \ \&\& \ (i>=i+1) \ \ (15>25-5)$	28	$(1==1) \ \&\& \ (0==0)$
29	$((2<=2) \ \&\& \ (3>=3)) \ \ (15>25)$	30	$(i<=i-1) \ \ !(7-10>0)$
31	$(1>10) \ \&\& \ (4>5)$	32	$(4*3==12) \ \ (7-5==0)$
33	$(1<10) \ \&\& \ (4<25)$	34	$((2<=2) \ \&\& \ (3>=3)) \ \ (15>25)$
35	$(2-2==9) \ \&\& \ !(3>=3)$	36	$(6<4) \ \&\& \ !(4>8)$
37	$(2*2==4) \ \ (4<2)$	38	$(A<=A+1) \ \ (12-3>0)$
39	$(i>=i-1) \ \ (3-5>0)$	40	$!(1<10) \ \ (12-3>0)$

Завдання 1b. Скласти таблицю істинності для логічного виразу

№ вар.	Вираз	№ вар.	Вираз
1	$a \parallel (!a \&\& b)$	7	$b \parallel (a \parallel b)$
2	$b \parallel (a \parallel !b)$	8	$a \parallel (a \&\& b)$
3	$b \&\& (!a \parallel b)$	9	$!a \&\& !b$
4	$b \&\& !(a \parallel b)$	10	$!a \parallel !b$
5	$!b \parallel a$	11	$b \&\& !a$
6	$a \&\& (a \parallel b)$	12	$!a \parallel b$
13	$!b \parallel (a \parallel b)$	14	$!a \parallel !(b \&\& a)$
15	$b \parallel !(a \&\& b)$	16	$(a \parallel b) \&\& !b$
17	$b \parallel (a \parallel b)$	24	$a \parallel (!a \&\& b)$
18	$a \parallel (a \&\& b)$	25	$b \parallel (a \parallel !b)$
19	$!a \&\& !b$	26	$b \&\& (!a \parallel b)$
20	$!a \parallel !b$	27	$b \&\& !(a \parallel b)$
21	$b \&\& !a$	28	$!b \parallel a$
22	$b \parallel !(a \&\& b)$	29	$a \&\& (a \parallel b)$
23	$!a \parallel !(b \&\& a)$	30	$!b \parallel (a \parallel b)$
31	$a \&\& (!a \parallel b)$	32	$a \parallel (!a \&\& b)$
33	$b \&\& (!a \parallel b)$	34	$b \parallel (a \parallel !b)$
35	$b \&\& !(a \parallel b)$	36	$a \parallel (a \&\& b)$
37	$!a \parallel !b$	38	$b \&\& !a$
39	$!b \parallel (a \parallel b)$	40	$b \parallel !(a \&\& b)$

Завдання 2. Обчислити і вивести на екран значення функції **y** для заданого користувачем значення змінної **x**. Скористайтесь прикладом 2.2

№ вар.	Завдання	№ вар.	Завдання
1	$y = \begin{cases} \ln(x), & x \geq 1 \\ \frac{\sqrt{x^2 + a^3}}{a}, & -1 < x < 1, \\ e^x, & x \leq -1 \end{cases}$ $a = 1,8$	2	$y = \begin{cases} \pi x^2, & x < 1,4 \\ ax^3 + \frac{8}{\sqrt{x}}, & x = 1,4 \\ \ln(x + 9\sqrt{ x + a }), & x > 1,4 \end{cases}$ $a = 1,65$
3	$y = \begin{cases} -x - 1, & x < -1 \\ 1 - x^2 \cos(\pi x), & -1 \leq x < 1, \\ x - \frac{1}{\sqrt{a^3}}, & x \geq 1 \end{cases}$ $a = 3,1$	4	$y = \begin{cases} e^{\frac{x^2}{2}} + \ln(a), & x < 1,2 \\ a \cos^3\left(\frac{\pi x}{2,7}\right), & x = 1,2 \\ x^2 + ax + 56, & x > 1,2 \end{cases}$ $a = 4$
5	$y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 1,3 \\ ax^3 + \frac{7}{\sqrt{x}}, & x = 1,3 \\ \tan(x + 9\sqrt{x}), & x > 1,3 \end{cases}$ $a = 1,5$	6	$y = \begin{cases} (x - 4)^2 + a, & x \leq 1 \\ 1,7 \cos^2(x), & 1 < x \leq 2 \\ e^{ax} \cos(bx), & x > 2 \end{cases}$ $a = 5,1, \\ b = 0,9$

№ вар.	Завдання	№ вар.	Завдання
7	$y = \begin{cases} 0, & x \leq 0 \\ x^2 - \frac{x}{a}, & 0 < x \leq 1 \\ x^3 - \sin(\pi x^2) - 1, & x > 1 \end{cases}$ $a = 1,8$	8	$y = \begin{cases} x^3 \sqrt{x-a}, & x > a \\ x \sin(ax), & x = a \\ e^{-a} \cos(ax), & x < a \end{cases}$ $a = 2,5,$
9	$y = \begin{cases} \sqrt{\frac{x^3}{x+a}} - \ln x, & x > 0 \\ 2a \ln(-x), & x < 0 \\ 0, & x = 0 \end{cases}$ $a = 0,7$	10	$y = \begin{cases} 1, & ax < 1 \\ ax^2 - \log(ax), & ax = 1 \\ ax^3 + \log(ax), & ax > 1 \end{cases}$ $a = 1,5$
11	$y = \begin{cases} ax^2 + bx, & x < 1,2 \\ \frac{a}{x} + \sqrt{x^2 + 1}, & x = 1,2 \\ \frac{a + bx}{\sqrt{x^2 + 1}}, & x > 1,2 \end{cases}$ $a = 2,8$	12	$y = \begin{cases} ax^2 - \log(ax), & x > 0 \\ \frac{a}{x} + \sqrt{x^2 + 1}, & x < 0 \\ 0, & x = 0 \end{cases}$ $a = 0,7$
13	$y = \begin{cases} ae^{-bx} \cos(\pi x), & x > 1 \\ (x-5)^{2^3} \sqrt{(x+1)^2}, & x \leq -1 \\ 0, & -1 < x \leq 1 \end{cases}$ $a = 0,7$	14	$y = \begin{cases} a + \cos^2(5,7x), & x < a \\ a + \sin^2 b^2, & a \leq x \leq 8 \\ a - \cos(x) , & x > 8 \end{cases}$ $a = 3,1$
15	$y = \begin{cases} \sin(bx) + \cos x^2, & x \leq a \\ \sqrt{a} + \sin x, & a < x < \ln(b), \\ \ln(a + bx), & x \geq \ln(b), \end{cases}$ $a = -1,2$ $b = 0,75$	16	$y = \begin{cases} (x-4)^2 + a, & x \leq 1 \\ 1,7 \cos^2 x^2, & a \leq x \leq 8 \\ e^{ax} \cos(bx), & x > 2 \end{cases}$ $a = 5,1$ $b = 0,9$
17	$y = \begin{cases} 3.5 \sin^2(bx + e^x)^3 \\ \ln(1.2 + b^3 x) + 1.2, \\ \cos^2(1.2 + x e^x) + b^2 \end{cases}$ $b = 2.5$	18	$y = \begin{cases} \sqrt{\frac{x^3}{x+a}} - \ln(x), & x > 0 \\ 2a \ln(-x), & x < 0 \\ 0, & x = 0 \end{cases}$ $a = 0,7$
19	$y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 1.3 \\ ax^3 + \frac{7}{\sqrt{x}}, & x = 1.3 \\ \operatorname{tg}(x + 7\sqrt{x}), & x > 1.3 \end{cases}$ $a = 1,5$	20	$y = \begin{cases} x^3 \sqrt{x-a}, & x > a \\ x \cos(ax), & x = a \\ e^{-ax} \sin(ax), & x < a \end{cases}$ $a = 2,5$
21	$y = \begin{cases} e^{\frac{x^2}{2}} + \ln(2a), & x < 1,2 \\ a \sin^3\left(\frac{\pi x}{2,7}\right), & x = 1,2 \\ x^2 + ax + 56, & x > 1,2 \end{cases}$ $a = 4,2$	22	$y = \begin{cases} 0, & x \leq 0 \\ x^2 - \frac{x}{a}, & 0 < x \leq 1 \\ x^3 - \sin(\pi x^2) - 1, & x > 1 \end{cases}$ $a = 1,8$

№ вар.	Завдання	№ вар.	Завдання
23	$= \begin{cases} 1, & ax < 1 \\ ax^2 - \log(ax), & ax = 1 \\ ax^3 + \log(ax), & ax > 1 \end{cases}$ $a = 1,5$	24	$y = \begin{cases} -x - 1, & x < -1 \\ 1 - x^2 \cos(\pi x), & -1 \leq x < 1 \\ x - \frac{1}{\sqrt{a^3}}, & x \geq 1 \end{cases},$ $a = 3,1$
25	$y = \begin{cases} ae^{-bx} \cos(\pi x), & x > 1 \\ (x-5)^2 \sqrt[3]{(x+1)^2}, & x \leq -1 \\ 0, & -1 < x \leq 1 \end{cases}$ $a = 0,7$	26	$y = \begin{cases} 3.5 \sin^2(bx + e^x)^3 \\ \ln(1.2 + b^3 x) + 1.2, \\ \cos^2(1.2 + xe^x) + b^2 \end{cases}$ $b = 2,5$
27	$y = \begin{cases} \sqrt{\frac{x^3}{x+a}} - \ln(x), & x > 0 \\ 2a \ln(-x), & x < 0 \\ 0, & x = 0 \end{cases}$ $a = 0,7$	28	$y = \begin{cases} x^3 \sqrt{x-a}, & x > a \\ x \sin(ax), & x = a \\ e^{-ax} \cos(ax), & x < a \end{cases}$ $a = 2,5,$
29	$y = \begin{cases} a + \cos^2(5,7x), & x < a \\ a + \sin^2 b^2, & a \leq x \leq 8 \\ a - \cos(x) , & x > 8 \end{cases}$ $a = 3,1$	30	$y = \begin{cases} 1, & ax < 1 \\ ax^2 - \log(ax), & ax = 1 \\ ax^3 + \log(ax), & ax > 1 \end{cases}$ $a = 1,5$
31	$y = \begin{cases} -x - 1, & x < -1 \\ 1 - x^2 \cos(\pi x), & -1 \leq x < 1 \\ x - \frac{1}{\sqrt{a^3}}, & x \geq 1 \end{cases}$ $a = 3,1$	32	$y = \begin{cases} ax - \cos^2 x, & x < a \\ (a)^2 - \ln(x), & 0,4 \leq x \leq 3,5a \\ \sqrt{8,1x + x^2}, & x > 3,5a \end{cases}$ $a = 0,4$
33	$y = \begin{cases} a + \cos^2(5,7x), & x < a \\ a + \sin^2(b^2), & a \leq x \leq 8 \\ a - \cos(x) , & x > 8 \end{cases}$ $a = 3,5$	34	$y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 1.3 \\ ax^3 + \frac{7}{\sqrt{x}}, & x = 1.3 \\ \operatorname{tg}(x + 7\sqrt{x}), & x > 1.3 \end{cases}$ $a = 1,5$
35	$y = \begin{cases} (x-4)^2 + a, & x \leq 1 \\ 1,7 \cos^2(x), & 1 < x \leq 2 \\ e^{ax} \cos(bx), & x > 2 \end{cases}$ $a = 5,1, b = 0,9$	36	$y = \begin{cases} x^3 \sqrt{x-a}, & x > a \\ x \sin(ax), & x = a \\ e^{-ax} \cos(ax), & x < a \end{cases}$ $a = 2,5,$
37	$y = \begin{cases} a^\pi \ln(bx) + \cos x^2, & x \leq a \\ \ln(x), & a < x < \ln(b), \\ \ln(a + bx), & x \geq \ln(b), \end{cases}$ $a = -1,2$ $b = 0,75$	38	$y = \begin{cases} 1, & ax < 1 \\ ax^2 - \log(ax), & ax = 1 \\ ax^3 + \log(ax), & ax > 1 \end{cases}$ $a = 1,5$

№ вар.	Завдання	№ вар.	Завдання
39	$y = \begin{cases} \sqrt{\frac{x^3}{x+a}} - \ln x, & x > 0 \\ 2a \ln(-x), & x < 0 \\ 0, & x = 0 \end{cases}$ $a = 0,7$	40	$y = \begin{cases} (x-4)^2 + a, & x \leq 1 \\ 1,7 \cos^2(x), & 1 < x \leq 2 \\ e^{ax} \cos(bx), & x > 2 \end{cases}$ $a = 5,1$ $b = 0,9$

Завдання 3. Написати програму обчислення функції **y (x)** для всіх різних варіантів вхідних параметрів. Ввести з клавіатури значення **x**, вивести значення функції для всіх варіантів (застосувати оператор switch, див. приклад 2.3).

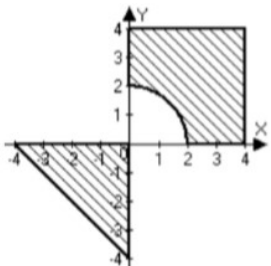
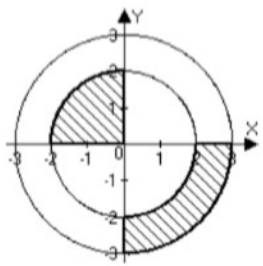
№ вар	Функція	Параметри
1	$y = \begin{cases} \frac{(2u+1)^2}{7\pi+x}, & \text{якщо } (u+x) > -0,5 \\ \cos^2 u - \sin \frac{u}{3}, & \text{якщо } -0,5 \leq u+x \leq 10^{-3} \\ \frac{\log(u+x) - e^x}{3,5x}, & \text{якщо } u+x \geq 10^{-3} \end{cases}$	1. $u = \sin(x)$ 2. $u = \cos(x)$ 3. $u = \operatorname{tg}(x)$
2	$y = \begin{cases} abx - \cos^2(zx), & \text{якщо } x < 3,5a \\ (a-x)^2 - \ln(z+x), & \text{якщо } 3,5a \leq x \leq b \\ \sqrt{bx-a+zx^2}, & \text{якщо } x > b \end{cases}$	1. $a = 0,4; b = 2,3; z = e^{2x}$ 2. $a = 0,2; b = 0,8; z = e^x$ 3. $a = 0,7; b = 8,1; z = 0,8$
3	$y = \begin{cases} \sin(bm + \cos(nx)), & \text{якщо } bm > x^2 \\ \cos(bm - \sin(x)), & \text{якщо } bm < x^2 \\ \sqrt{e^{ \cos x } + \sqrt{ bmx }}, & \text{якщо } bm = x^2 \end{cases}$	1. $b = -1,6; m = 0,9; n = -1,4$ 2. $b = 4,5; m = -2; n = 2,2$ 3. $b = -4,5; m = 0,5; n = -1,5$
4	$y = \begin{cases} a \sin^2(x) + b \cos(zx), & \text{якщо } x < -\ln(a) \\ a^b - \cos^3(a+zx), & \text{якщо } -\ln(a) \leq x \leq b \\ \sqrt{2,5a^3 + (b-zx^2)^6}, & \text{якщо } x > b \end{cases}$	1. $a = 0,2; b = 0,5; z = e^{ax}$ 2. $a = 0,15; b = 0,2; z = e^{2ax}$ 3. $a = 0,9; b = 5; z = e^{2,5ax}$
5	$y = \begin{cases} \sin(xe^{a+b}) + x^2, & \text{якщо } e^{a+b} > e^x \\ a^b - \cos^3(a+zx), & \text{якщо } e^{a+b} = e^x \\ \sqrt{2,5a^3 + (b-zx^2)^6}, & \text{якщо } e^{a+b} < e^x \end{cases}$	1. $a = 4,2; b = 5,3; c = 1,5$ 2. $a = -0,35; b = 1,8; c = -1,8$ 3. $a = 2,8; b = -0,6; c = 2,0$
6	$y = \begin{cases} 2,8 \sin^2(ax) + bx^3z, & \text{якщо } x < a \\ z \cos(ax+b)^2 + \ln(z), & \text{якщо } a \leq x \leq b^2 \\ e^{2,5ax} + zabx, & \text{якщо } x > b^2 \end{cases}$	1. $a = -5; b = 2,5; z = \ln bx^3 $ 2. $a = 3; b = 5; z = \ln bx $ 3. $a = -10; b = 3; z = \ln bx^2 $
7	$y = \begin{cases} xe^a + e^{ bc }, & \text{якщо } 1-x^2 = a+c \\ \sin^2 ax + \cos bc, & \text{якщо } 1-x^2 > a+c \\ \sqrt{ab^4 + \sqrt[5]{cx^2}}, & \text{якщо } 1-x^2 < a+c \end{cases}$	1. $a = 3,2; b = -0,7; c = 2,2$ 2. $a = 10,5; b = -2,5; c = 5,6$ 3. $a = 5,4; b = 3; c = 2,6$
8	$y = \begin{cases} \ln mx+n , & \text{якщо } x^2 > m+n \\ e^{\cos mx-n }, & \text{якщо } x^2 = m+n \\ \sqrt[3]{k^2 + \cos^2(x)}, & \text{якщо } x^2 < m+n \end{cases}$	1. $k = 3,1; m = 5,15; n = -1,15$ 2. $k = 0,78; m = -2,4; n = 4,36$ 3. $k = 1,1; m = 0,8; n = 0,41$

№ вар	Функція	Параметри
9	$y = \begin{cases} a \sin^2(x) + b \cos(zx + a), & \text{якщо } x < a^3 \\ (a + bx)^2 - \sin(a + zx), & \text{якщо } a^3 \leq x \leq b \\ \sqrt{x - \sin(bx + z)}, & \text{якщо } x > b \end{cases}$	1. $a = 1,2; b = 7,2; z = e^x$ 2. $a = -1,5; b = 3,2; z = e^{2x}$ 3. $a = 1,7; b = 5,5; z = e^3$
10	$y = \begin{cases} \sqrt[3]{b^2 + \sqrt{ x + c }}, & \text{якщо } \log(a) < x \\ \cos(x - b - c), & \text{якщо } \log(a) = x \\ \sin(x + a - b), & \text{якщо } \log(a) > x \end{cases}$	1. $a = 0,1; b = 9,8; c = 11,12$ 2. $a = 10; b = 10,05; c = 6,2$ 3. $a = 100; b = 3,03; c = 7,12$
11	$y = \begin{cases} e^{ax} - 3,5 \cos^2(z + bx), & \text{якщо } x \leq a \\ a + \ln a + bx - 2x, & \text{якщо } a < x < b^{3,5} \\ a + \cos^{3,5}(a + bxz), & \text{якщо } x > b^{3,5} \end{cases}$	1. $a = -1; b = 3,4; z = \operatorname{tg}(bx)$ 2. $a = -3,2; b = 5,5; z = \operatorname{tg}(bx^2)$ 3. $a = -5,2; b = 7,2; z = \operatorname{tg}(bx^3)$
12	$y = \begin{cases} \ln(\log kx + mn), & \text{якщо } 3x > m + n \\ \sin(kmx) + \sqrt{ nx }, & \text{якщо } 3x = m + n \\ e^{\cos(x)} + e^{m+n}, & \text{якщо } 3x < m + n \end{cases}$	1. $k = 4; m = -14,7; n = -0,6$ 2. $k = 3; m = 6,5; n = 3,15$ 3. $k = 5; m = -12; n = 0,45$
13	$y = \begin{cases} x^2 e^{2k} + \ln rx , & \text{якщо } \cos(x) = \cos(rs) \\ \sqrt[3]{x^2} + \sqrt{ k + rsx }, & \text{якщо } \cos(x) > \cos(rs) \\ \operatorname{arctg}(kx + rs), & \text{якщо } \cos(x) < \cos(rs) \end{cases}$	1. $k = 1,33; r = 0,85; s = 3,5$ 2. $k = 0,9; r = 3,3; s = 1,2$ 3. $k = 1,57; r = 0,75; s = 2,15$
14	$y = \begin{cases} 2,5b^2 + ax - 4,5 \cos(xz), & \text{якщо } x \leq 5a \\ (a^2 - 5,4x)^3 + \ln(xz), & \text{якщо } x > b \\ \sqrt{6,5b^2 + (a - x^3z)}, & \text{якщо } 5a < x \leq b \end{cases}$	1. $a = 0,5; b = 4,5; z = e^{ax}$ 2. $a = 0,5; b = 3,7; z = e^{2ax}$ 3. $a = 0,5; b = 2,7; z = e^{2,5ax}$
15	$y = \begin{cases} \sqrt{ ax - \cos^2(b^3x) + 5,1c^2 }, & \text{якщо } 1 - x^2 = a + c \\ e^{0,004x} + \ln b^5 \cos(x) , & \text{якщо } 1 - x^2 > a + c \\ \cos^2(b^3x^2) + \ln bx - a^2 , & \text{якщо } 1 - x^2 < a + c \end{cases}$	1. $a = 3,5; b = -0,73; c = 2,5$ 2. $a = 15,4; b = -5,6; c = 3,5$ 3. $a = 5,1; b = 4; c = 2,7$
16	$y = \begin{cases} 3,5 \sin^2(bx + z)^3 - e^{3,5a}, & \text{якщо } x \leq a \\ \ln(a + b^3x) + a, & \text{якщо } a < x \leq b^{2,5} \\ \cos^2(a^b + xz) + a^2, & \text{якщо } x > b^{2,5} \end{cases}$	1. $a = 0,1; b = 0,5; z = e^{2,5ax}$ 2. $a = 1,2; b = 2,5; z = e^{2,5ax}$ 3. $a = 2,5; b = 1,2; z = e^{2,5ax}$
17	$y = \begin{cases} a + \sin(bx) + \cos(x^2), & \text{якщо } x \leq a \\ \sqrt{a + bx} + \sin(zx), & \text{якщо } a < x < \ln(b) \\ \ln(a + bx + z), & \text{якщо } x \geq \ln(b) \end{cases}$	1. $a = -1,2; b = 0,7; z = \ln(\operatorname{tg}(bx))$ 2. $a = 0,4; b = 2,4; z = \ln(\operatorname{tg}(bx))$ 3. $a = 1,1; b = 6,1; z = \ln(\operatorname{tg}(bx))$
18	$y = \begin{cases} (3,5a - 7,3bx + \sin(zx))^3, & \text{якщо } x < -\ln a \\ a^b - \cos^3(a + zx), & \text{якщо } -\ln a \leq x < b \\ \sqrt{ \operatorname{tg}(a) - x } - x^2, & \text{якщо } x > b \end{cases}$	1. $a = 6; b = 3,2; z = e^{1,5ax}$ 2. $a = 3; b = 6; z = e^{1,5ax}$ 3. $a = 2,7; b = 1,8; z = e^{1,5ax}$
19	$y = \begin{cases} c \sin(b^2x) + b \ln(cx + a), & \text{якщо } x < a \\ a + \ln(bx) - \sin^2(a + cx), & \text{якщо } a \leq x < b \\ \sqrt{ \cos(a + bx) + cx^2 }, & \text{якщо } x \geq b \end{cases}$	1. $a = 2,2; b = 2,4; c = \ln bx $ 2. $a = 1,6; b = 1,7; c = \ln bx $ 3. $a = 1,3; b = 4,2; c = \ln b^2x $
20	$y = \begin{cases} e^{ax} + f \cos^5(bx), & \text{якщо } x \leq a \\ a + \cos^2(bx) - \ln(fx), & \text{якщо } a < x \leq b^2 \\ \cos^2(a + bfx), & \text{якщо } x > b^2 \end{cases}$	1. $a = 0,8; b = 2,4; f = e^{1,5ax}$ 2. $a = 1,2; b = 4,2; f = e^{2ax}$ 3. $a = 3,4; b = 8,1; f = e^{3ax}$
21	$y = \begin{cases} a \cos^2(x) + b \sin(zx), & \text{якщо } x \leq a \\ a \operatorname{tg}(ax + z) + \sin^2(bx), & \text{якщо } a < x \leq 4,5b \\ \ln(ax - b) + z^2, & \text{якщо } x > 4,5b \end{cases}$	1. $a = 4,5; b = 8,4; z = \operatorname{tg}(bx)^2$ 2. $a = 8,2; b = 15,2; z = \operatorname{tg}(bx)^2$ 3. $a = 1,7; b = 0,5; z = \operatorname{tg}(bx)^2$

№ вар	Функція	Параметри
22	$y = \begin{cases} a + bx + \sin^2(zx^{3,5}), & \text{якщо } x < a \\ a + \ln ab - zx^3 + \ln(x), & \text{якщо } a \leq x \leq b^2 \\ \sqrt{ a + \operatorname{ctg}(zx) } + b\sin(x), & \text{якщо } x > b^2 \end{cases}$	1. $a = 0,3; b = 0,9; z = \sin(x^2)$ 2. $a = 4,3; b = 3,15; z = \sin(x^3)$ 3. $a = 1,7; b = 0,5; z = \sin^2(x)$
23	$y = \begin{cases} \ln bzx + za^{2,5}, & \text{якщо } a^3 < x \leq b \\ ax^2 + bz^a + \sin^2(zx), & \text{якщо } x > b \\ \cos(ax + b) + \ln zx , & \text{якщо } x \leq a^3 \end{cases}$	1. $a = 1,5; b = 6,4; z = \ln bx^3 $ 2. $a = 1,9; b = 8,6; z = \ln bx^3 $ 3. $a = 0,6; b = 2,4; z = \ln bx^3 $
24	$y = \begin{cases} xe^x + (z + 7,7abx), & \text{якщо } x < a \\ \operatorname{tg}(ax + z) + \cos^2(bx), & \text{якщо } a \leq x \leq b^2 \\ \ln(\sin^2(a + bx + zx^2)), & \text{якщо } x > b^2 \end{cases}$	1. $a = 8,7; b = 3,7; z = \operatorname{tg}(bx)$ 2. $a = 9,3; b = 3,5; z = \operatorname{tg}(abx)$ 3. $a = 2,1; b = 5,7; z = \operatorname{tg}(b^2x)$
25	$y = \begin{cases} \sqrt[3]{0,3b + \sqrt{ (a - z^2 - \cos(x)) }}, & \text{якщо } x > b \\ a + z\cos^2(bx)^3, & \text{якщо } x < a \\ a + \sin^2(b) + \ln(zx), & \text{якщо } a \leq x \leq b \end{cases}$	1. $a = 1,5; b = 5,7; z = \ln(\operatorname{tg}(bx))$ 2. $a = 3,7; b = 8,4; z = \ln(\operatorname{tg}(bx))$ 3. $a = 4,4; b = 5,6; z = \ln(\operatorname{tg}(bx))$
26	$y = \begin{cases} \frac{(2z + 1)^2}{3,71 - x^2}, & \text{якщо } z > -0,5 \\ \sin^3(z) - \sin \frac{z}{3\pi}, & \text{якщо } -0,5 \leq z \leq 10^{-3} \\ \frac{\operatorname{tg}(z + x) - e^x}{3,5x}, & \text{якщо } z \geq 10^{-3} \end{cases}$	1. $z = \arcsin(x^3)$ 2. $z = \arccos^2(x)$ 3. $z = \operatorname{tg}(x)$
27	$y = \begin{cases} a^2x^3 + \sqrt{b^4 + 1,7}, & \text{якщо } x < 0,2 \\ \operatorname{arctg}(2^x - p), & \text{якщо } x = 0,2 \\ \sqrt[3]{\ln(a) + 4,3 + x}, & \text{якщо } x > 0,2 \end{cases}$	1. $a = 0,5; b = 1,5; p = -4$ 2. $a = -1; b = 0,5; p = -4$ 3. $a = -2; b = 0; p = -4$
28	$y = \begin{cases} \sin(e^{a+b}) + x^2, & \text{якщо } a + b > x \\ \operatorname{arctg}(abc) + \sqrt[3]{x}, & \text{якщо } a + b = x \\ \arcsin(\cos^2(\sqrt{ x })), & \text{якщо } a + b < x \end{cases}$	1. $a = 7,2; b = -1,3; c = 2,5$ 2. $a = 1,47; b = 3,81; c = 2,8$ 3. $a = 4,8; b = 10,6; c = 2,7$
29	$y = \begin{cases} \operatorname{ctg}(x^2e^{3k}) + \ln r + x , & \text{якщо } x = rs \\ \sqrt[5]{x^2} + \sqrt{ \arcsin(k) }, & \text{якщо } x > rs \\ \operatorname{arctg}(kx + \operatorname{tg}(rs)), & \text{якщо } x < rs \end{cases}$	1. $k = -0,3; r = 0,85; s = 3,5$ 2. $k = 0,9; r = 3,3; s = 1,2$ 3. $k = -0,7; r = 0,75; s = 2,15$
30	$y = \begin{cases} a^3 + \operatorname{arctg}(\sin^3(bx)) + \cos^2(x^2), & \text{якщо } x \leq 2 \\ \sqrt{(a + bx) + 2} + \sin(zx), & \text{якщо } a < x < \ln b \\ \operatorname{arctg}(a + bx + z), & \text{якщо } x > 0,2 \end{cases}$	1. $a = 1,5; b = 5,7; z = \operatorname{tg} \operatorname{tg}(bx) $ 2. $a = 3,7; b = 8,4; z = \operatorname{tg} \operatorname{tg}(bx) $ 3. $a = 4,4; b = 5,6; z = \operatorname{tg} \operatorname{tg}(bx) $
31	$y = \begin{cases} \frac{(2u + 1)^2}{7\pi + x}, & \text{якщо } (u + x) > -0,5 \\ \cos^2 u - \sin \frac{u}{3}, & \text{якщо } -0,5 \leq (u + x) \leq 10^{-3} \\ \frac{\log(u + x) - e^x}{3,5x}, & \text{якщо } (u + x) \geq 10^{-3} \end{cases}$	1. $u = \sin(x)$ 2. $u = \cos(x)$ 3. $u = \operatorname{tg}(x)$
32	$y = \begin{cases} abx - \cos^2(zx), & \text{якщо } x < 3,5a \\ (a - x)^2 - \ln(z + x), & \text{якщо } 3,5a \leq x \leq b \\ \sqrt{bx - a + zx^2}, & \text{якщо } x > b \end{cases}$	1. $a = 0,4; b = 2,3; z = e^{2x}$ 2. $a = 0,2; b = 0,8; z = e^x$ 3. $a = 0,7; b = 8,1; z = 0,8$

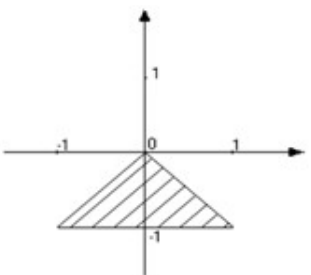
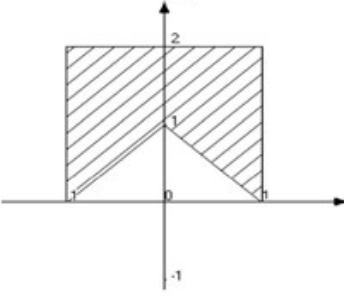
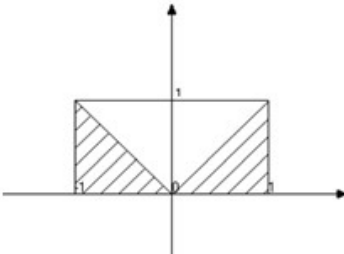
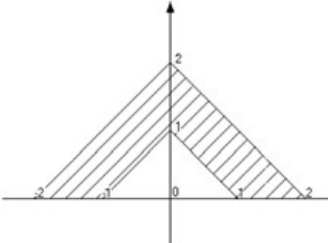
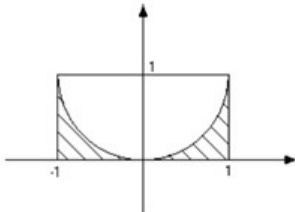
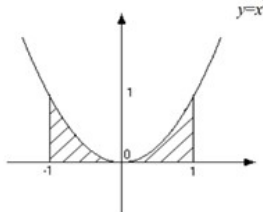
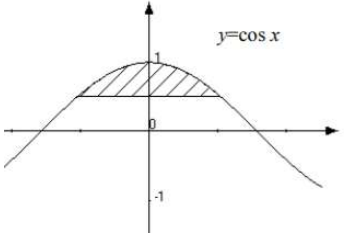
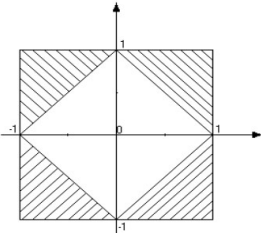
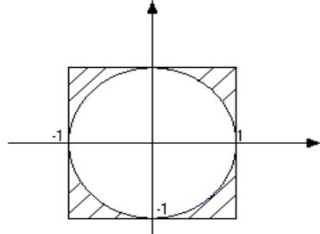
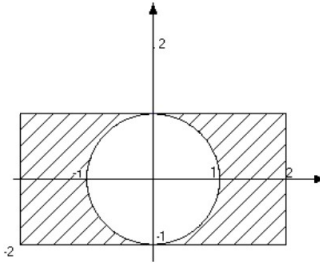
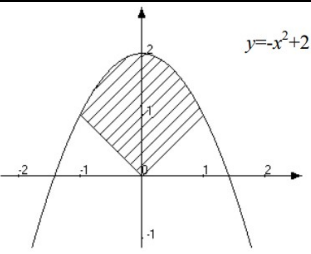
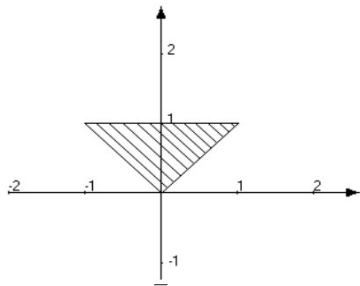
№ вар	Функція	Параметри
33	$y = \begin{cases} \sin(bm + \cos(nx)), & \text{якщо } bm > x^2 \\ \cos(bm - \sin(x)), & \text{якщо } bm < x^2 \\ \sqrt{e^{ \cos x }} + \sqrt{ bmx }, & \text{якщо } bm = x^2 \end{cases}$	1. $b = -1,6; m = 0,9; n = -1,4$ 2. $b = 4,5; m = -2; n = 2,2$ 3. $b = -4,5; m = 0,5; n = -1,5$
34	$y = \begin{cases} a \sin^2(x) + b \cos(zx), & \text{якщо } x < -\ln(a) \\ a^b - \cos^3(a + zx), & \text{якщо } -\ln(a) \leq x \leq b \\ \sqrt{2,5a^3 + (b - zx^2)^6}, & \text{якщо } x > b \end{cases}$	1. $a = 0,2; b = 0,5; z = e^{ax}$ 2. $a = 0,15; b = 0,2; z = e^{2ax}$ 3. $a = 0,9; b = 5; z = e^{2,5ax}$
35	$y = \begin{cases} \sin(e^{a+b}) + x^2, & \text{якщо } e^{a+b} > e^x \\ \arctg(abc) + \sqrt[3]{x}, & \text{якщо } e^{a+b} = e^x \\ \cos(\sqrt{ x + abc }), & \text{якщо } e^{a+b} < e^x \end{cases}$	1. $a = 4,2; b = 5,3; c = 1,5$ 2. $a = -0,35; b = 1,8; c = -1,8$ 3. $a = 2,8; b = -0,6; c = 2,0$
36	$y = \begin{cases} 2,8 \sin^2(ax) - bx^3z, & \text{якщо } x < a \\ z \cos(ax + b)^2 + \ln(z), & \text{якщо } a \leq x \leq b^2 \\ e^{2,5ax} + zabx, & \text{якщо } x > b^2 \end{cases}$	1. $a = -5; b = 2,5; z = \ln bx^3 $ 2. $a = 3; b = 5; z = \ln bx $ 3. $a = 10; b = 2,5; z = \ln bx^2 $
37	$y = \begin{cases} xe^a + e^{ bc }, & \text{якщо } 1 - x^2 = a + c \\ \sin^2 ax + \cos bc, & \text{якщо } 1 - x^2 > a + c \\ \sqrt{ab^4 + \sqrt[5]{cx^2}}, & \text{якщо } 1 - x^2 < a + c \end{cases}$	1. $a = 3,2; b = -0,7; c = 2,2$ 2. $a = 10,5; b = -2,5; c = 5,6$ 3. $a = 5,4; b = 3; c = 2,6$
38	$y = \begin{cases} \ln mx + n , & \text{якщо } x^2 > m + n \\ e^{\cos mx-n }, & \text{якщо } x^2 = m + n \\ \sqrt[3]{x^2 + \cos^2 x}, & \text{якщо } x^2 < m + n \end{cases}$	1. $k = 3,1; m = 5,15; n = -1,15$ 2. $k = 0,78; m = -2,4; n = 4,36$ 3. $k = 1,1; m = 0,8; n = 0,41$
39	$y = \begin{cases} a \sin^2(x) + b \cos(zx + a), & \text{якщо } x < a^3 \\ (a + bx)^2 - \sin(a + zx), & \text{якщо } a^3 \leq x \leq b \\ e^{2,5ax} + zabx, & \text{якщо } x > b \end{cases}$	1. $a = 1,2; b = 7,2; z = e^x$ 2. $a = -1,5; b = 3,2; z = e^{2x}$ 3. $a = 1,7; b = 5,5; z = e^3$
40	$y = \begin{cases} \sqrt[3]{b^2 + \sqrt{ x + c }}, & \text{якщо } \lg a < x \\ \cos(x - b - c), & \text{якщо } \lg a = x \\ \sin(x + a - b), & \text{якщо } \lg a > x \end{cases}$	1. $a = 0,1; b = 9,8; c = 11,12$ 2. $a = 10; b = 10,05; c = 6,2$ 3. $a = 100; b = 3,03; c = 7,12$

Завдання 4. За заданими координатами (x, y) точки визначити, чи належить ця точка заштрихованій області. Координати точки ввести з клавіатури. Скористайтесь [довідковими матеріалами \(див. нижче\)](#).

№ вар	Область	№ вар.	Область
1		2	

№ вар	Область	№ вар.	Область
3		4	
5		6	
7		8	
9		10	
11		12	
13		14	

№ вар	Область	№ вар.	Область
15		16	
17		18	
19		20	
21		22	
23		24	

№ вар	Область	№ вар.	Область
25		26	
27		28	
29		30	
31		32	
33		34	
35		36	

№ вар	Область	№ вар.	Область
37		38	
39		40	

Довідкові матеріали до виконання завдання №4

1. Точка $A(x, y)$ належить заштрихованій області (див. рис. 2.3), якщо її координати задовольняють такій нерівності: $(x - x_0)^2 + (y - y_0)^2 \leq R^2$,
 x, y – координати точки A ,
 x_0, y_0 – координати центру круга
 R – величина радіуса круга

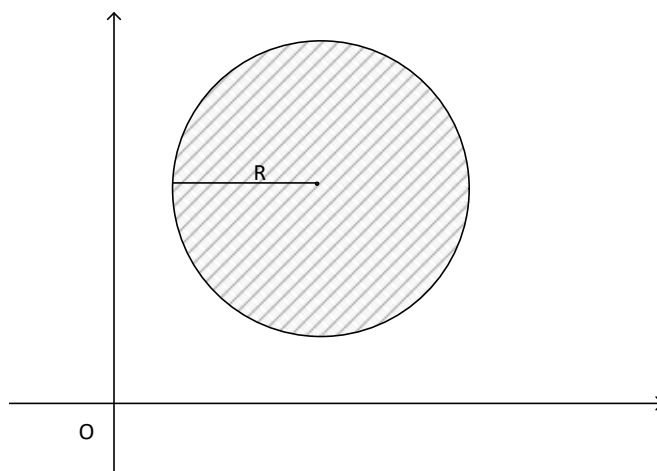


Рисунок 2.3 – Точка A розташована всередині круга

Відповідна умова в тексті програми матиме такий вигляд:

```
if ((pow(A_x - x_0, 2) + pow(A_y - y_0, 2)) <= R * R,
```

де A_x, A_y – координати точки A , x_0, y_0 – координати центру круга.

2. Точка $A(x, y)$ належить заштрихованій області (див. рис. 2.4), якщо її координати задовольняють такій нерівності $(x - x_0)^2 + (y - y_0)^2 \geq R^2$,
 x, y – координати точки A ,
 x_0, y_0 – координати центру круга
 R – величина радіуса круга

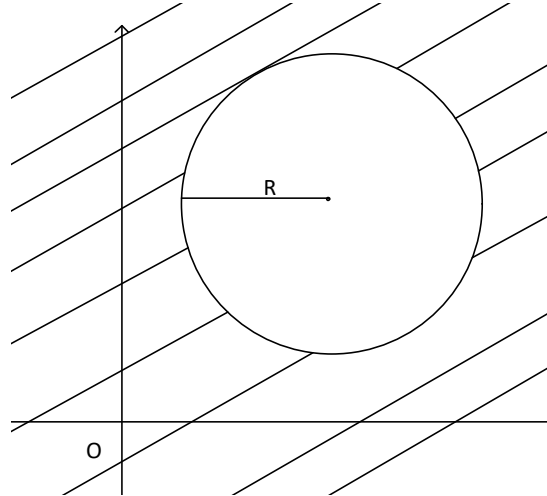


Рисунок 2.4 – Точка A знаходиться за межами кола

Відповідна умова в тексті програми матиме такий вигляд:

```
if ((pow(A_x - x_0, 2) + pow(A_y - y_0, 2)) >= R * R),
```

де A_x, A_y – координати точки A , x_0, y_0 – координати центру круга

3. Записати систему нерівностей для координат точки $A(A_x, A_y)$, щоб вона знаходилась всередині заштрихованої області.

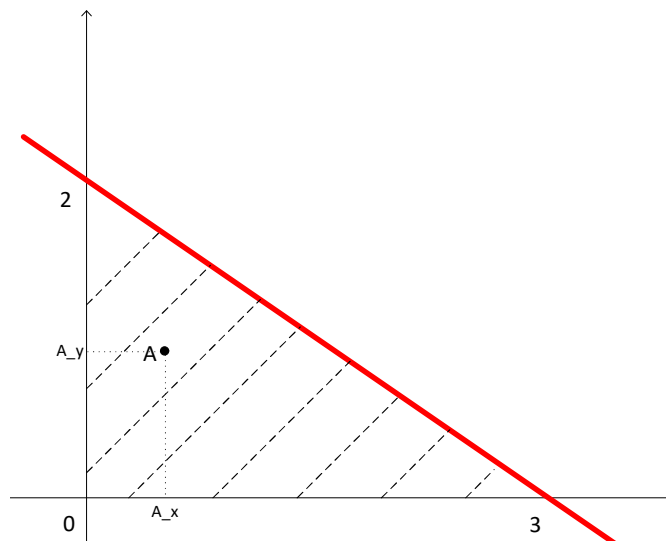


Рисунок 2.5 – Точка знаходиться всередині трикутника

Рівняння прямої, позначеної червоним має вид:

$$y = -\frac{2}{3}x + 2$$

Отже, для того, щоб точка A потрапила в область її координати мають задовольняти такій нерівності (підставляємо координати точки у рівняння прямої):

$$A_y \leq -\frac{2}{3} \cdot A_x + 2$$

Крім цього, заштрихована область обмежена ще й координатними вісями. Отже, координати точки A задовольнятимуть таким додатковим обмеженням:

$$\begin{cases} 0 \leq A_x \leq 3 \\ 0 \leq A_y \leq 2 \end{cases}$$

Таким чином, маємо таку систему обмежень для координат точки A в заштрихованій області:

$$\begin{cases} A_y \leq -\frac{2}{3} \cdot A_x + 2 \\ 0 \leq A_x \leq 3 \\ 0 \leq A_y \leq 2 \end{cases}$$

Тут виникає питання. Яким чином таку систему обмежень записати в програмі? А от тут доречно буде використати логічні операції, які було розглянуто раніше.

Перша нерівність виглядатиме так:

$$A_y \leq (-2./3 * A_x + 2)$$

Друга і третя нерівності відповідно:

$$(A_x \geq 0) \ \&\& \ (A_x \leq 3)$$

$$(A_y \geq 0) \ \&\& \ (A_y \leq 2)$$

зверніть увагу на логічну операцію **&&** (логічне **І**), якою замінено подвійну нерівність.

Нарешті, в програмі весь блок нерівностей матиме такий вигляд:

```
if ((A_y <= (-2./3 * A_x + 2)) && (A_x >= 0) && (A_x <= 3) &&
&& (A_y >= 0) && (A_y <= 2))
```

Якщо ж таких заштрихованих областей декілька, то для «зв'язки» систем обмежень між областями можна використати логічну операцію **АБО**. Альтернативний варіант: для кожної області можна створити окремий **if**.

Контрольні запитання

1. Який пріоритет виконання логічних операцій?
2. Чи можливо застосовувати логічні операції до цілих операндів?
3. Які існують варіанти умовного оператора?
4. Що таке оператор безумовного переходу? В чому його небезпека при використанні в коді?
5. Для чого використовується оператор **case**?
6. Що означає термін «магічні числа»? В чому полягає небезпека використання таких чисел в програмі?

Корисні посилання

1. Оператори умовного розгалуження **if/else**. URL:
<https://acode.com.ua/urok-67-operator-umovnogo-rozgaluzhennya-if-else/>
2. Оператор **switch**. URL : <https://acode.com.ua/urok-68-operator-switch/>
3. Умовний тернарний оператор, оператор **sizeof** і Кома. URL:
<https://acode.com.ua/urok-44-umovnyj-ternarnyj-operator-sizeof-i-koma/>
4. Відлагодження програм: степпінг і точки зупину. URL:
<https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
5. C++ Programming Style Guidelines. URL:
<https://geosoft.no/development/cppstyle.html>

Лабораторне заняття № 3. Програмування циклічних алгоритмів

Мета: ознайомитись з видами циклів у мові C та набутти практичних навичок програмування циклічних алгоритмів.

План

1. Цикли з параметром.
2. Цикл з передумовою.
3. Цикл з післяумовою.
4. Завдання до лабораторної роботи.
5. Контрольні запитання.
6. Корисні посилання.

3.1 Цикли з параметром

В заголовку циклу з параметром явним чином задається кількість повторень тіла циклу, а параметр циклу змінюється автоматично всередині тіла циклу.

Цикли **for** зазвичай використовують, коли заздалегідь відомо кількість повторів виконання певної інструкції.

Приклад 3.1 Обчислити значення виразу (даний вираз читається так: сума чисел від 1 до 10)

$$S = \sum_{i=1}^{10} i$$

Блок-схема алгоритму розв'язання цієї задачі має вид:

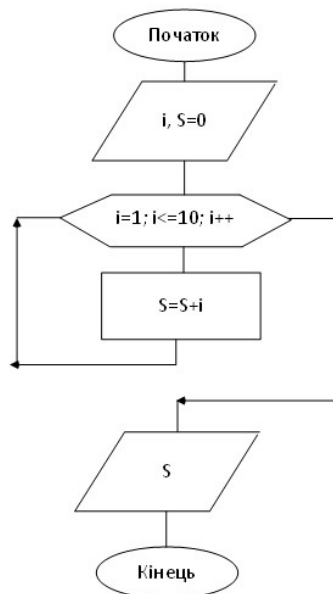


Рисунок 3.1 – Блок-схема до задачі про обчислення суми ряду

Один з варіантів програмної реалізації наведено на рисунку 3.2

```
#include <stdio.h>

int main()
{
    int i, S = 0, number = 10;
    for (i = 1; i <= number; i++)
    {
        S += i; //аналогічно запису S = S+i
    }
    printf("Suma of numbers is equal %d", S);
}
```

Рисунок 3.2 – Лістинг програми обчислення суми ряду

Далі в таблиці наведено поетапний процес виконання програми:

Значення лічильника	Перевірка умови	Виконання операторів в тілі цикла Значення суми	Оновлення лічильника
i = 1	1<=10 - умова істинна, виконуються оператори в тілі цикла	S = S+i=0+1 = 1	i =i+1 = 1+1 = 2
i = 2	2<=10 - умова істинна	S = S+i= 1+2 = 3	i =i+1 = 2+1 = 3
i = 3	3<=10 - умова істинна	S = S+i= 3+3 = 6	i =i+1 = 3+1 = 4
i = 4	4<=10 - умова істинна	S =S+i= 6+4 = 10	i =i+1 = 4+1 = 5
i = 5	5<=10 - умова істинна	S =S+i= 10+5 = 15	i =i+1 = 5+1 = 6
i = 6	6<=10 - умова істинна	S =S+i=15+6 = 21	i =i+1 = 6+1 = 7
i = 7	7<=10 - умова істинна	S =S+i=21+7 = 28	i =i+1 = 7+1 = 8
i = 8	8<=10 - умова істинна	S =S+i=28+8 = 36	i =i+1 = 8+1 = 9
i = 9	9<=10 - умова істинна	S =S+i=36+9 = 45	i=i+1=9+1 = 10
i = 10	10<=10 - умова істинна	S=S+i=45+10 = 55	i=i+1= 10+1 = 11
i = 11	11<=10 - умова не є істинною. Виконання циклу переривається!!!		

Тип змінної **i** можна вказати як до циклу, так і всередині конструкції **for**. Тобто зробити такий запис **for** (int i = 1; i <= number; i++)

Декілька коментарів до програми. Запис **i++** означає, що змінна циклу (тобто лічильник) на кожному етапі роботи циклу змінюється з кроком 1. Окрім цього зверніть увагу на ініціалізацію нулем змінної **S**, яка накопичуватиме суму на кожній ітерації. Для чого це зроблено? На початку роботи програми, до ініціалізації, в комірці пам'яті з ім'ям **S** може зберігатись будь що.

Нижче на скріншоті у вікні налагодження програми можна побачити, які значення зберігаються в комірках **i** та **S** до ініціалізації (**-858993460**)

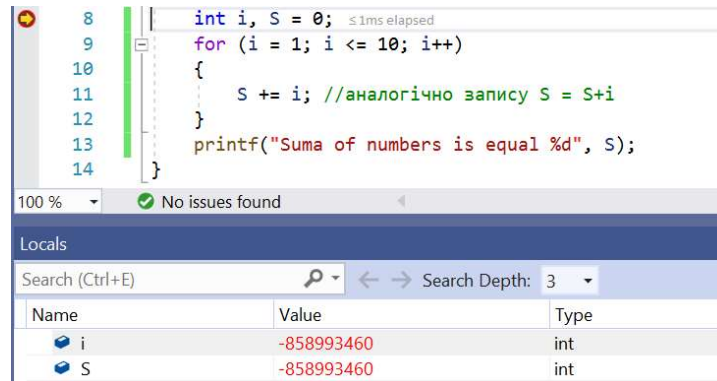


Рисунок 3.3 – Вікно налагодження програми

Приклад 3.2 Протабулювати функцію $y = |\log_{10}(x)| - (x - 2)^2$ на проміжку $[x_1; x_2]$

Програмна реалізація даної задачі наведена на рисунку 3.4. Нижче буде наведено коментарі до основних етапів роботи програми.

```

4  #include <stdio.h>
5  #include <math.h>
6
7  int main()
8  {
9      float y;
10     int x1 = -10, x2 = 10; //задаємо інтервал значень аргументу
11     int count = 1;
12     printf("\t*****Function tab*****\n\n");
13     for (x1; x1 <= x2; x1++) {
14         if (x1 > 0) {
15             y = abs(log10(x1)) - powf((x1 - 2), 2);
16             printf("|%4d.| x = %3d | y = %8.4f | \n", count, x1, y);
17             count++;
18         }
19         else {
20             printf("|%4d.| x = %3d | y = No value | \n", count, x1);
21             count++;
22         }
23     }
24     printf("\n\t*****Final*****\n");
25     return 1;
26 }

```

Рисунок 3.4 – Лістинг програми табулювання функції

Декілька коментарів до цієї програми. Інтервал значень аргументу задаємо безпосередньо в програмі. Звісно, початкові і кінцеві значення аргументів можна задати з клавіатури. Певну кількість пробілів в **printf** додано для гарного виведення на екран результатів. Змінну **count** введено для відображення номеру рядка в таблиці. В рядку № 16 параметр **8.4** означає, що 8 – це загальна кількість знакомісць, яке відводиться під значення змінної, 4 – кількість позицій для дробної частини (тобто після десяткової крапки). У цьому прикладі кількість знакомісць у виводимому числі менша за 8, тому вільні місця зліва від числа заповнюються пробілами. Такий спосіб форматування доцільно використовувати для побудови таблиць. Результат програми наведено на рисунку 3.5.

```
*****Function tab*****
```

1.	x = -10	y = No value
2.	x = -9	y = No value
3.	x = -8	y = No value
4.	x = -7	y = No value
5.	x = -6	y = No value
6.	x = -5	y = No value
7.	x = -4	y = No value
8.	x = -3	y = No value
9.	x = -2	y = No value
10.	x = -1	y = No value
11.	x = 0	y = No value
12.	x = 1	y = -1.0000
13.	x = 2	y = 0.0000
14.	x = 3	y = -1.0000
15.	x = 4	y = -4.0000
16.	x = 5	y = -9.0000
17.	x = 6	y = -16.0000
18.	x = 7	y = -25.0000
19.	x = 8	y = -36.0000
20.	x = 9	y = -49.0000
21.	x = 10	y = -63.0000

```
*****Final*****
```

Рисунок 3.5 – Таблиця значень функції

При використанні вкладених циклів потрібно слідкувати, щоб внутрішній цикл повністю вкладався у тіло зовнішнього циклу. Крім того, внутрішній цикл може містити власні вкладені цикли. Обмеження на кількість вкладених циклів немає. Імена параметрів зовнішнього та внутрішнього циклів повинні бути різними.

Приклад 3.3 Написати програму, яка друкує на екрані таблицю Піфагора

```
#include <stdio.h>
int main()
{
    int i, j;
    printf("      ");
    for (j = 1; j <= 10; j++) {
        printf("%4d", j);
    }
    printf("\n");
    for (i = 1; i <= 10; i++) {
        printf("%4d", i);
        for (j = 1; j <= 10; j++) {
            printf("%4d", i * j);
        }
    }
}
```

```

    }
    printf("\n");
}
printf("\nClick ENTER to finish");

```

Рисунок 3.6 – Лістинг програми «Таблиця Піфагора»

3.2 Цикл з передумовою

Цикли **while** використовуються тоді, коли кількість ітерацій (повторень) заздалегідь невідома. Натомість, є умова, істинність якої перевіряється при кожному проході циклу. Якщо умова набуває значення «Істина» – виконуємо тіло циклу ще раз, якщо «Хиба» – виходимо з нього.

Приклад 3.4 Вивести на екран числа від 1 до 10 у зворотньому порядку:

```

int count = 10;
while (count >= 0)
{
    printf("%i\n", count);
    count--;
}

```

Рисунок 3.7 – Програма друку чисел в зворотньому порядку

Програма виконуватиметься до тих пір, доки змінна **count** не набуде значення **-1**. Таким чином, на екрані буде надруковано числа від **1** до **10**. Зменшення значення змінної **count** на одиницю на кожній ітерації циклу забезпечуватиме друк в зворотньому порядку.

3.3 Цикл з постумовою

Циклічний оператор **do-while** спочатку виконує дію, потім перевіряє умову. Якщо умова істинна, то повторює дію, знову перевіряє умову і так далі.

Цикл **do-while**, на відміну від **while**, обов'язково виконується хоча б один раз, навіть якщо умова хибна. Зазвичай цикл з постумовою застосовується тоді, коли необхідно виконати цикл хоча б один раз, наприклад, при очікуванні введення від користувача.

Приклад 3.5 Програма запитує введення числа у користувача, допоки він не введе додатне число:

```

int input;
do{
    printf("Enter a positive number: ");
    input = GetInt();
}
while(input < 0);

```

Рисунок 3.8 – Програмна реалізація циклу з постумовою

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №3

Відповідно до варіанта (визначається за списком академгрупи) виконати 5 завдань. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1-5;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта.
3. Умови завдань 1-5, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №3

Завдання №1. Розробити блок-схему та написати програму для табулювання функції, тобто вивести всі значення функції у заданому діапазоні з вказаним кроком **0,2**. Інтервал значень змінної **x** наведено в таблиці. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями. Подбайте про гарне виведення на екран таблиці. Самостійно обрати оператор циклу.

№ варіанта	Функція	Інтервал
1.	$y = x^2 + \sin 5x$	[0.1;2.1]
2.	$y = x^2 - \cos^2 \pi x$	[1;2]
3.	$y = 1.8x^2 - \sin 10x$	[0.2;2.2]
4.	$y = 2^x - 2x^2 - 1$	[2;4]
5.	$y = x^2 - \cos^2(x + 1)$	[0.1;2.2]
6.	$y = x^3 - 4x^2 + 2$	[1;3]
7.	$y = x - 3 \cos^2(1.04x)$	[0.15;2.1]
8.	$y = e^{2x} - 2 - x^2$	[1;3]
9.	$y = x^3 + 3x^2 - 3$	[2;4]
10.	$y = x^2 + 2\pi \cos(\pi x)$	[0.1;2.5]
11.	$y = 5x^3 + 2x^2 - 15x - 6$	[1.3;3.4]
12.	$y = \log_{10} x - (x - 2)^2$	[2;4.1]
13.	$y = \sqrt{x} - 2 \cos(0.5\pi x)$	[0.1;2.5]
14.	$y = x^2 - x\pi \cos \pi x$	[0.1;2.1]
15.	$y = x^3 - 7x - 7$	[1;3]
16.	$y = x^2 - 4 + 0.25x - 2$	[0.1;2.7]
17.	$y = x^2 - \sin \pi x$	[0.1;2.9]
18.	$y = \ln x^2 - x + 4$	[1;2.9]
19.	$y = x^3 - 6x^2 + 2$	[1;2.9]
20.	$y = 3\sin\sqrt{x} + 0.25x - 3$	[1;3]
21.	$y = 3 \cos x - x - 4 + 4$	[0;2]

№ варіанта	Функція	Інтервал
22.	$y = 0.25x^3 - 2.8x - 2$	[1.1;2.9]
23.	$y = \ln x^2 - 1.8 \sin x$	[1;3]
24.	$y = x^2 + 4 \sin \pi x$	[0.1;3]
25.	$y = 0.5x^2 - 1 - \lg(x - 3)$	[1;2.9]
26.	$y = \sqrt{ 1 - x } - 3 \cos x$	[0.1;3]
27.	$y = \ln x^2 + x - 5$	[1;3]
28.	$y = x^3 - 1.75x^2 + 0.75$	[1;3]
29.	$y = 0.5x - 1 - 2 \cos(x + \frac{\pi}{4})$	[0;2]
30.	$y = \pi^2 - \cos 3x$	[0.8;3.9]
31.	$y = 3x - 2 \ln x^2 - 5$	[2;4]
32.	$y = x^2 + \sin 5x$	[1.1;3.1]
33.	$y = x^2 - \cos^2 \pi x$	[3;4]
34.	$y = 1.8x^2 - \sin 10x$	[1.2;3.2]
35.	$y = 2^x - 2x^2 - 1$	[4;6]
36.	$y = x^2 - \cos^2(x + 1)$	[1.1;3.2]
37.	$y = x^3 - 4x^2 + 2$	[2;4]
38.	$y = x^2 - 3 \cos^2(1.04x)$	[0.25;2.15]
39.	$y = e^{2x} - 2 - x^2$	[0.5;2.5]
40.	$y = x^3 + 3x^2 - 3$	[1;3]

Завдання №2. Скласти блок-схему алгоритму та програму обчислення суми n членів ряду згідно умові задачі. Значення x ввести з клавіатури. Кількість елементів ряду (n) користувач вводить з клавіатури. Обов'язковою умовою є виведення на екран номера ітерації та суми на кожному кроці обчислення. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями. Використайте цикл з параметром (**for**)

№ вар.	Сума S	№ вар.	Сума S
1	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$	17	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$
2	$\cos x + \frac{\cos 2x}{2} + \frac{\cos 3x}{3} + \dots$	18	$\frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \frac{\cos 6x}{35} + \dots$
3	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$	19	$1 + \frac{2x}{1!} + \frac{(2x)^2}{2!} + \dots$
4	$\sin x - \frac{\sin 2x}{2} + \frac{\sin 3x}{3} - \dots$	20	$1 + \frac{2}{1!} \left(\frac{x}{2}\right) + \frac{5}{2!} \left(\frac{x}{2}\right)^2 + \frac{10}{3!} \left(\frac{x}{2}\right)^3 + \dots$
5	$\cos x + \frac{\cos 3x}{9} + \frac{\cos 5x}{25} + \dots$	21	$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$
6	$1 + \frac{\cos \frac{\pi}{4}}{1!} x + \frac{\cos 2 \frac{\pi}{4}}{2!} x^2 + \dots$	22	$1 - \frac{3}{2!} x^2 + \frac{5}{4!} x^4 - \frac{10}{6!} x^6 + \dots$
7	$x - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$	23	$-\frac{(2x)^2}{2!} + \frac{(2x)^4}{4!} - \frac{(2x)^6}{6!} + \dots$

№ вар.	Сума S	№ вар.	Сума S
8	$x \sin(\frac{\pi}{4}) + x^2 \sin(2\frac{\pi}{4}) + \dots$	24	$-(x+1)^2 + \frac{(x+1)^4}{2} - \frac{(x+1)^6}{3} + \dots$
9	$x + \frac{x^5}{5!} + \frac{x^9}{9!} + \frac{x^{13}}{13!}$	25	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$
10	$1 + \frac{\cos x}{1!} + \frac{\cos 2x}{2!} + \dots$	26	$\frac{x \sin(\frac{\pi}{3})}{1} + \frac{x^2 \sin(2\frac{\pi}{3})}{2} + \dots$
11	$1 + \frac{3x^2}{1!} + \frac{5x^4}{2!} + \dots$	27	$1 + \frac{\sin(\frac{\pi}{4})}{1!} + \frac{\sin(2\frac{\pi}{4})}{2!} x^2 + \dots$
12	$\frac{x \cos(\frac{\pi}{3})}{1} + \frac{x^2 \cos(2\frac{\pi}{3})}{2} + \dots$	28	$x \cos(\frac{\pi}{4}) + x^2 \cos 2\frac{\pi}{4} + \dots$
13	$\frac{x-1}{x+2} + \frac{1}{3}(\frac{x-1}{x+1})^3 + \frac{1}{5}(\frac{x-1}{x+1})^5 + \dots$	29	$\frac{1}{2}(\frac{1}{x+1})^2 + \frac{1}{4}(\frac{1}{x+1})^4 + \dots$
14	$-\cos x + \frac{\cos 2x}{4} - \frac{\cos 3x}{9} + \dots$	30	$\sin(x) + \sin(3x^3) + \sin(5x^5) + \dots$
15	$\frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} - \dots$	31	$1 - \frac{x^2}{2} + \frac{x^4}{4} - \frac{x^6}{6} + \dots$
16	$\sin x + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots$	32	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$
33	$\cos x + \frac{\cos 2x}{2} + \frac{\cos 3x}{3} + \dots$	34	$\sin x - \frac{\sin 2x}{2} + \frac{\sin 3x}{3} - \dots$
35	$1 + \frac{2}{1!}(\frac{x}{2}) + \frac{5}{2!}(\frac{x}{2})^2 + \frac{10}{3!}(\frac{x}{2})^3 + \dots$	36	$\cos x + \frac{\cos 3x}{9} + \frac{\cos 5x}{25} + \dots$
37	$x \sin(\frac{\pi}{4}) + x^2 \sin(2\frac{\pi}{4}) + \dots$	38	$\frac{x \cos(\frac{\pi}{3})}{1} + \frac{x^2 \cos(2\frac{\pi}{3})}{2} + \dots$
39	$1 + \frac{\cos x}{1!} + \frac{\cos 2x}{2!} + \dots$	40	$1 + \frac{3x^2}{1!} + \frac{5x^4}{2!} + \dots$

Завдання №3 Написати програму для обчислення значення суми. Кількість членів ряду (число k) вводить користувач з клавіатури. Обов'язковою умовою є виведення на екран номера ітерації та суми на кожному кроці обчислення. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями. **В цій програмі реалізувати всі три типи циклів.**

Зверніть увагу! Результати обчислень всіх трьох типів циклів повинні співпадати

№ вар.	Сума S	№ вар.	Сума S
1.	$\sum_{k=1}^{\infty} \frac{5}{k^3 + 2}$	16	$\sum_{k=1}^{\infty} \frac{2^k}{k(2^k + 1)}$
2.	$\sum_{k=1}^{\infty} \frac{k^3}{3^k}$	17	$\sum_{k=1}^{\infty} \frac{7}{k^2 + 4}$

№ вар.	Сума S	№ вар.	Сума S
3.	$\sum_{k=1}^{\infty} \frac{1}{k^2}$	18	$\sum_{k=1}^{\infty} \frac{1}{4^k}$
4.	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^4}$	19	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3+2}$
5.	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3-k^2+1}$	20	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3+1}$
6.	$\sum_{k=1}^{\infty} (-1)^k \frac{2k+1}{k^3+2}$	21	$\sum_{k=1}^{\infty} (-1)^k \frac{k-1}{2k^2+1}$
7.	$\sum_{k=1}^{\infty} \frac{\cos(2k-1)}{(2k-1)^2}$	22	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k+1}}{(2k+1)(2k-1)}$
8.	$\sum_{k=1}^{\infty} \frac{x^k \cdot \sin(\frac{2\pi k}{3})}{k}$	23	$\sum_{k=1}^{\infty} \frac{5}{k^3+2}$
9.	$\sum_{k=1}^{\infty} \frac{2^k}{k(2^k+1)}$	24	$\sum_{k=1}^{\infty} \frac{k^3}{3^k}$
10.	$\sum_{k=1}^{\infty} \frac{7}{k^2+4}$	25	$\sum_{k=1}^{\infty} \frac{1}{k^2}$
11.	$\sum_{k=1}^{\infty} \frac{1}{4^k}$	26	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^4}$
12.	$\sum_{k=1}^{\infty} (-1)^k \frac{3k-1}{k^3+1}$	27	$\sum_{k=1}^{\infty} (-1)^k \frac{k-1}{k^3+1}$
13.	$\sum_{k=1}^{\infty} (-1)^k \frac{3k+1}{3k^3-1}$	28	$\sum_{k=1}^{\infty} \frac{\sin(k-1)}{(2k-1)}$
14.	$\sum_{k=1}^{\infty} (-1)^k \frac{k+4}{k^2+4}$	29	$\sum_{k=1}^{\infty} \frac{\cos(2k-1)}{(2k-1)^2}$
15.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k+1}}{(2k+1)(2k-1)}$	30	$\sum_{k=1}^{\infty} \frac{x^k \cdot \sin(\frac{2\pi k}{3})}{k}$
32	$k + \sum_{k=1}^{\infty} \frac{1}{k}$	31	$\sum_{i=0}^{\infty} \frac{1}{2^i}$
33	$\sum_{k=1}^{\infty} \left(\frac{1}{k} + 1\right)^2$	34	$\sum_{k=1}^{\infty} (-1)^k \frac{3k+1}{k^2-1}$
35	$\sum_{k=1}^{\infty} \frac{k}{k^2-1}$	36	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{2k+1}{(2k-1)^2}$
37	$\sum_{k=1}^{\infty} \frac{(k+1)^3}{3^k}$	38	$\sum_{k=1}^{\infty} \frac{k}{k^3+1}$

№ вар.	Сума S	№ вар.	Сума S
39	$\sum_{k=1}^{\infty} \frac{(k-1)^3}{3^k}$	40	$\sum_{k=1}^{\infty} \frac{1}{k^3}$

Завдання №4. Розробити блок-схему та програму для обчислення суми ряду. Сумування виконується до того моменту, поки член ряду більший за значення похибки ϵ (вводить користувач з клавіатури). Обов'язковою умовою є вивід на екран номеру та суми на кожному кроці обчислення.

Примітка. Скористайтесь результатом попередньої програми для оцінки похибки, адже це число має бути досить невеликим (наприклад, 0,0007). Використати оператор **while** або **do-while**

№ вар.	Сума S	№ вар.	Сума S
1.	$\sum_{k=1}^{\infty} \frac{5}{k^3 + 2}$	16	$\sum_{k=1}^{\infty} \frac{2^k}{k(2^k + 1)}$
2.	$\sum_{k=1}^{\infty} \frac{k^3}{3^k}$	17	$\sum_{k=1}^{\infty} \frac{7}{k^2 + 4}$
3.	$\sum_{k=1}^{\infty} \frac{1}{k^2}$	18	$\sum_{k=1}^{\infty} \frac{1}{4^k}$
4.	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^4}$	19	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3 + 2}$
5.	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3 - k^2 + 1}$	20	$\sum_{k=1}^{\infty} (-1)^k \frac{k+1}{k^3 + 1}$
6.	$\sum_{k=1}^{\infty} (-1)^k \frac{2k+1}{k^3 + 2}$	21	$\sum_{k=1}^{\infty} (-1)^k \frac{k-1}{2k^2 + 1}$
7.	$\sum_{k=1}^{\infty} \frac{\cos(2k-1)}{(2k-1)^2}$	22	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k+1}}{(2k+1)(2k-1)}$
8.	$\sum_{k=1}^{\infty} \frac{x^k \cdot \sin(\frac{2\pi k}{3})}{k}$	23	$\sum_{k=1}^{\infty} \frac{5}{k^3 + 2}$
9.	$\sum_{k=1}^{\infty} \frac{2^k}{k(2^k + 1)}$	24	$\sum_{k=1}^{\infty} \frac{k^3}{3^k}$
10.	$\sum_{k=1}^{\infty} \frac{7}{k^2 + 4}$	25	$\sum_{k=1}^{\infty} \frac{1}{k^2}$
11.	$\sum_{k=1}^{\infty} \frac{1}{4^k}$	26	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^4}$
12.	$\sum_{k=1}^{\infty} (-1)^k \frac{3k-1}{k^3 + 1}$	27	$\sum_{k=1}^{\infty} (-1)^k \frac{k-1}{k^3 + 1}$

№ вар.	Сума S	№ вар.	Сума S
13.	$\sum_{k=1}^{\infty} (-1)^k \frac{3k+1}{3k^3-1}$	28	$\sum_{k=1}^{\infty} \frac{\sin(k-1)}{(2k-1)}$
14.	$\sum_{k=1}^{\infty} (-1)^k \frac{k+4}{k^2+4}$	29	$\sum_{k=1}^{\infty} \frac{\cos(2k-1)}{(2k-1)^2}$
15.	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k+1}}{(2k+1)(2k-1)}$	30	$\sum_{k=1}^{\infty} \frac{x^k \cdot \sin(\frac{2\pi k}{3})}{k}$
32	$k + \sum_{k=1}^{\infty} \frac{1}{k}$	31	$\sum_{i=0}^{\infty} \frac{1}{2^i}$
33	$\sum_{k=1}^{\infty} \left(\frac{1}{k} + 1\right)^2$	34	$\sum_{k=1}^{\infty} (-1)^k \frac{3k+1}{k^2-1}$
35	$\sum_{k=1}^{\infty} \frac{k}{k^2-1}$	36	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{2k+1}{(2k-1)^2}$
37	$\sum_{k=1}^{\infty} \frac{(k+1)^3}{3^k}$	38	$\sum_{k=1}^{\infty} \frac{k}{k^3+1}$
39	$\sum_{k=1}^{\infty} \frac{(k-1)^3}{3^k}$	40	$\sum_{k=1}^{\infty} \frac{1}{k^3}$

Результат, орієнтовно, виглядатиме так:

```

Microsoft Visual Studio Debug Console
Enter value of epsilon
0.0007
1) y=0.25000000 s=0.250000
2) y=0.06250000 s=0.312500
3) y=0.01562500 s=0.328125
4) y=0.00390625 s=0.332031
5) y=0.00097656 s=0.333008

y = 0.000244 is less than mistake = 0.000700
That's why we stop computation
This is the sum of 5 members s=0.333008

```

Рисунок 3.9 – Відповідь до завдання 4

Завдання №5. Розробити блок-схему та програму для вирішення задачі. Тип циклу обрати самостійно з огляду на доцільність. Вхідні дані вводити з клавіатури

№ вар.	Завдання
1	Піднести число A до цілого степеня N . Використовувати функцію pow заборонено.
2	Обчислити факторіал заданого числа. Передбачити відповідний тип для результату.
3	Обчислити суму квадратів чисел від 1 до N . На екран послідовно виводити число, його квадрат та суму отриману на кожному кроці обчислення.
4	Обчислити суму кубів непарних чисел від 1 до N . На екран послідовно виводити

№ вар.	Завдання
	число, його куб та суму отриману на кожному кроці обчислення.
5	Знайти всі числа, що є кратними п'яти для чисел від I до N (ввести з клавіатури).
6	Чи є задане натуральне число степенем двійки?
7	Розкласти задане число на прості множники. Вивести інформацію на екран. Окремо надрукувати кількість таких множників.
8	Надрукувати таблицю переведення відстаней з дюймів в сантиметри для значень довжин від 1 до 20 дюймів. 1 дюйм=2,54 см
9	Знайти суму цілих від'ємних чисел, що є кратними 5. Інтервал для дослідження ввести з клавіатури. Також вивести на екран кількість таких чисел.
10	З діапазона від 1 до 100 надрукувати всі числа, які діляться на три і не діляться на п'ять. Окремо вивести на екран кількість таких чисел.
11	Знайти суму цілих додатніх чисел, що є кратними 4 і меншими 100. На екран послідовно виводити число, суму отриману на кожному кроці обчислення.
12	Складіть програму, що обчислює різницю квадратів всіх чисел від I до N
13	Напишіть програму для виведення на екран всіх непарних чисел від 100 до 200 включно
14	Знайти квадрат різниці чисел від m до n . Необхідні дані ввести з клавіатури
15	Дано цілі числа m і n . Вивести на друк всі дільники числа m , що є взаємно простими з n
16	Дано натуральне число n в діапазоні від 100 до 800. Визначити кількість тризначних натуральних чисел, що є більшими за n
17	Знайти суму цілих чисел, що є кратними 9 в діапазоні від -100 до 100
18	Вивести на екран всі числа кратні п'яти в інтервалі від 2 до 100 включно
19	Вивести на екран всі тризначні числа від -500 до 500, у яких парні сотні. Окремо вивести на екран кількість таких чисел
20	Визначити суму модулів всіх непарних двозначних чисел. Окремо вивести на екран кількість таких чисел
21	Знайти всі тризначні числа в діапазоні від 0 до 700, у яких непарні сотні. Окремо вивести на екран кількість таких чисел
22	Надрукувати у порядку спадання всі дільники даного числа. Окремо вивести кількість таких дільників
23	Складіть програму, яка підраховуватиме кількість цифр числа, яке вводить користувач з клавіатури
24	Знайти всі числа, що не є кратними 11 для чисел від I до N (вводить користувач з клавіатури). Вивести на екран такі числа, та їх кількість.
25	Знайти суму цілих додатніх чисел, що є кратними 4 і меншими 100. Вивести на екран суму та кількість таких чисел
26	Перевести число з десяткової системи числення у вісімкову. Використовувати алгоритм ділення на 8
27	Напишіть програму для виведення на екран всіх парних чисел, що є кратними трьом, в діапазоні від 100 до 200 включно
28	Вивести на екран всі числа, що є кратними числам 2, 5, 14 з діапазону від 1 до 2000.
29	Знайти суму цілих додатніх чисел, що є кратними 8 і меншими 190
30	Чи є задане натуральне число степенем трійки?
31	Спортсмен у перший день пробіг 10 км. Кожен наступний день він збільшував норму на 10% від результату попереднього дня. Вивести довжину марафону, який пробіжить спортсмен на N день (N користувач вводить з клавіатури)
32	Обчислити суму квадратів чисел від I до N (ввести з клавіатури). На екран послідовно виводити число, його квадрат та суму отриману на цьому кроці

№ вар.	Завдання
	обчислення.
33	Обчислити суму кубів непарних чисел від 1 до N (ввести з клавіатури). На екран послідовно виводити число, його куб та суму отриману на цьому кроці обчислення.
34	Знайти всі числа, що є кратними п'яти для чисел від 1 до N (ввести з клавіатури).
35	Чи є задане (ввести з клавіатури) натуральне число степенем двійки?
36	Розкласти задане число на прості множники. Вивести інформацію на екран. Окремо надрукувати кількість таких множників.
37	Надрукувати таблицю переведення відстаней з дюймів в сантиметри для значень довжин від 1 до 20 дюймів. $1 \text{ дюйм} = 2,54 \text{ см}$
38	Знайти суму цілих від'ємних чисел, що є кратними 5 . Інтервал для дослідження ввести з клавіатури. Також вивести на екран кількість таких чисел.
39	З діапазона від 1 до 100 надрукувати всі числа, які діляться на три і не діляться на п'ять. Окремо вивести на екран кількість таких чисел.
40	Знайти суму цілих додатніх чисел, що є кратними 4 і меншими 100 . На екран послідовно виводити число, суму отриману на кожному кроці обчислення.

Контрольні запитання

1. Чи можливий зворотній відлік у операторі циклу із заданим числом повторень?
2. Який цикл завжди виконується хоча б один раз?
3. Яким чином створити нескінченний цикл?
4. Наведіть приклад циклу, який жодного разу не виконає тіло циклу.
5. Яким чином можна достроково вийти з циклу?
6. Чи можна при пошуку максимального значення у довільній послідовності чисел перше значення максимуму задавати рівним нулю? Чому?

Корисні посилання

1. Цикл **for**. URL: <https://acode.com.ua/urok-72-tsykl-for/>
2. Цикл **while**. URL: <https://acode.com.ua/urok-70-tsykl-while/>
3. Цикл **do-while**. URL: <https://acode.com.ua/urok-71-tsykl-do-while/>
4. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
5. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html>

Лабораторне заняття № 4 Обробка масивів

Мета: ознайомитись з правилами обробки масивів у мові C та набути практичних навичок обробки масивів у програмах.

План

1. Одновимірні масиви.
2. Двовимірні масиви.
3. Завдання до лабораторної роботи.
4. Довідкові матеріали до виконання завдання №3.
5. Контрольні запитання.
6. Корисні посилання.

4.1 Одновимірні масиви

Якщо елементи масива мають **один індекс**, то масив називається **одновимірним або лінійним, або масив-вектор**.

Для оголошення одновимірного масиву необхідно вказати: тип значення його елементів, ім'я масиву, кількість елементів масиву (див. рис. 4.1).

```
int nums[10] = { 1, 2, 3, 4 }  
int arr[] = { 1, 2, 3, 4, 5, 23, 34 }  
int cards[4] = { 1, 2, 3, 4 }
```

Рисунок 4.1 – Оголошення та ініціалізація одновимірного масиву

Наведемо приклад обробки елементів масиву в програмі (див. рис. 4.2). Для визначення кількості елементів масиву буде використано оператор **sizeof**.

```
#include <stdio.h>  
  
int main()  
{  
    int arr[] = { 1, 2, 3, 4, 5, 23, 34, 16, -3, 0 };  
    int N = sizeof arr/sizeof(int);  
    for (int i = 0; i < N; i++) {  
        printf(" %d ", arr[i]);  
    }  
}
```

Рисунок 4.2 – Приклад програми для виведення масиву на друк

В цій програмі довжина масиву **arr** в байтах обчислюється за допомогою функції **sizeof**. Для обчислення кількості елементів масиву необхідно розділити довжину масиву в байтах на кількість байт, які виділяються для типу даних **int**. Такий підхід надає можливість «не прив'язуватись» до конкретної

кількості елементів масиву та динамічно змінювати дане число в залежності від умов завдання.

Приклад 4.1 Ввести з клавіатури сім елементів масиву, обчислити середнє арифметичне елементів масиву.

```
#include <stdio.h>
int main()
{
    int const number = 7; //кількість елементів масиву
    int testArray[number]; //оголошення цілочисельного масиву
    //ініціалізація змінної для збереження суми елементів
    int sumOfArray = 0;
    //оголошення змінної для середнього значення
    double meanOfArray;
    printf("\tPlease enter the array elements\n");
    for (int i = 0; i < number; i++) {
        printf("array[%d] = ", i);
        scanf_s("%d", &testArray[i]);
    }
    printf("\n now we print our new array\n");
    for (int i = 0; i < number; i++) {
        printf("%4d", testArray[i]);
        sumOfArray = sumOfArray + testArray[i];
    }
    meanOfArray = sumOfArray / number;
    printf("\nthe mean value is: %lf", meanOfArray);}
```

Рисунок 4.3 – Обчислення середнього арифметичного елементів масиву

Зверніть увагу, що за необхідності обробити всі або декілька (наприклад, тільки ті, які задовольняють певній умові) елементів масиву необхідно застосовувати циклічний оператор. Також в даній програмі для досягнення необхідної точності потрібно врахувати тип даних, відповідно до якого визначається результат обчислення.

Приклад 4.2 Згенерувати масив псевдовипадкових чисел та обчислити кількість додатніх елементів в ньому.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int arr[20] = { 0 }; //ініціалізуємо нулем елементи масиву
    int lenghtOfArray;
    int positive; //лічильник для додатніх елементів
    printf("Enter number of elements. It may be less than 20");
    scanf_s("%d", &lenghtOfArray);
    //генеруємо випадкові числа в діапазоні від -20 до 10
    srand(time(NULL));
    for (int i = 0; i < lenghtOfArray; i++) {
```

```

arr[i] = rand() % (10+20+1)-20;
printf("%4d ", arr[i]);
}
printf("\n\n");
//обчислимо кількість додатніх елементів
for (int i = 0; i < lenghtOfArray; i++) {
    if (arr[i] > 0)
        positive++;
}
printf("\nNumber of positive elements is %d\n", positive);

return 0;
}

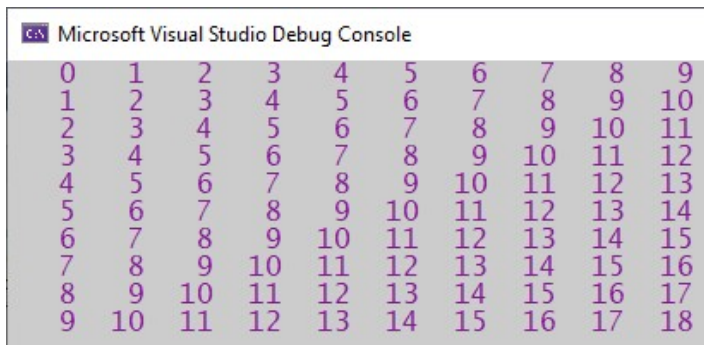
```

Рисунок 4.4 – Обчислення кількості додатніх елементів масиву

На рисунку 4.4 змінна **positive** виконує роль лічильника, який збільшується на 1 при виконанні певних умов. В даному випадку умова пов'язує збільшення лічильника на 1 з додатнім елементом, який відповідає поточному значенню лічильника циклу.

4.2 Двовимірні масиви

В деяких випадках зручно елементи масиву пронумерувати двома незалежними індексами, такі масиви називають двовимірними або матрицями. Значення елементів таких масивів зазвичай виводять на екран у вигляді таблиці (див. рис. 4.5).



Microsoft Visual Studio Debug Console

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	11	12
4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14
6	7	8	9	10	11	12	13	14	15
7	8	9	10	11	12	13	14	15	16
8	9	10	11	12	13	14	15	16	17
9	10	11	12	13	14	15	16	17	18

Рисунок 4.5 – Виведення двовимірного масиву на екран

Доступ до елементів здійснюватиметься за двома індексами, наприклад так **A[i][j]**, де **A** – ім'я масиву. Нижче на рисунку 4.6 наведено декілька способів оголошення та ініціалізації двовимірного масиву.

```

int a[2][3] = { 1, 2, 3, 4, 5, 6 };
int a[2][3];
int a[2][3] = { {1, 2, 3}, {4, 5, 6} };
int a[][3] = { {1, 2, 3}, {4, 5, 6} };

```

Рисунок 4.6 – Оголошення та ініціалізація двовимірного масиву

Оскільки масив двовимірний, то для його обробки необхідно використовувати конструкцію із вкладених циклічних операторів. Розглянемо відповідний приклад

Приклад 4.3 Згенерувати двовимірний масив із псевдовипадкових чисел та обчислити середнє арифметичне тих елементів, сума індексів яких є парним числом. Масив надрукувати у вигляді таблиці.

```
int main()
{
    int a[20][20] = { 0 };
    int N;
    int A, B, count = 0;
    float S = 0;
    printf("Enter interval for array generation\n");
    scanf_s("%d %d", &A, &B);
    printf("Enter size of array\n");
    scanf_s("%d", &N);
    srand(time(NULL));
    // масив заповнюємо випадковими числами з інтервалу [A; B]
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            a[i][j] = rand() % (B - A + 1) + A;
            printf("%4d", a[i][j]);
        }
        printf("\n"); //для переведення курсора на наступний рядок
        // і коректного друку двовимірного масиву
    }
    printf("Elements, that satisfy the condition...\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            if ((i + j) % 2 == 0) {
                count++;
                S += a[i][j];
            }
        }
    }
    if (count != 0) {
        printf("\nNumber of elements is %d , arithmetic mean=%f",
count, S/count);
    }
    else printf("\nthere are no elements that satisfy the
condition\n");
    return 0;
}
```

Рисунок 4.7 – Обчислення середнього арифметичного двовимірного масиву

Обробка двовимірного масиву здійснюється у вкладеній циклічній структурі, оскільки необхідно «пробігти» і за індексом рядка, і за індексом стовпця. Доступ до кожного елемента здійснюється за обома індексами.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №4

Відповідно до варіанта (визначається за списком академгрупи) виконати 3 завдання. В кожній програмі реалізувати нескінченний цикл. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1-3;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта
3. Умови завдань 1-3, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №4

Завдання 1. Одновимірні цілочисельні масиви. Всі варіанти завдань передбачають генерацію масиву випадкових чисел у діапазоні $[-20;20]$. Кількість елементів масиву вводиться з клавіатури користувачем, але максимальна довжина масиву складає 20 елементів. **Використовувати додатковий одновимірний масив забороняється.** Обов'язково виводити на екран згенерований масив. Передбачити в програмі можливість введення з клавіатури розмірності масиву та діапазону для генерування елементів масиву.

№ вар.	Завдання
1	Інвертувати масив, тобто поміняти місцями 1-й елемент з останнім, 2-й з передостаннім тощо. Як результат отримати початковий масив, записаний у зворотному порядку.
2	Обчислити суму від'ємних значень масиву та максимальний додатний елемент.
3	Змістити всі елементи масиву вправо на одну позицію. Обчислити максимум.
4	Обчислити добуток всіх парних елементів масиву та мінімального з непарних.
5	Обчислити середнє, мінімальне та максимальне значення всіх елементів масиву.
6	Замінити всі елементи, які кратні трьом, нулями.
7	Знайти суму елементів, що розташовані до першого від'ємного елемента. Якщо від'ємних елементів немає – надрукувати відповідне повідомлення
8	Визначити різницю між добутком всіх додатних елементів та модулів всіх від'ємних
9	Розмістити елементи масиву у порядку спадання. Визначити кількість перестановок, які було здійснено під час сортування.
10	Змістити всі елементи масиву вліво на одну позицію. Обчислити середнє арифметичне значень масиву.
11	Визначити суму мінімального елемента та його індекса.
12	Обчислити середнє геометричне та максимум всіх додатних елементів масиву.
13	Обчислити кількість елементів масиву, які є більшими за середнє значення масиву.
14	Обчислити кількість від'ємних елементів масиву, які є меншими середнього

№ вар.	Завдання
	значення масиву.
15	Обчислити середнє геометричне та максимальне значення модулів всіх елементів масиву.
16	Обчислити середнє геометричне та мінімальне значення абсолютних значень всіх від'ємних елементів.
17	Координати n -вимірного вектору задані у масиві (n є будь-яким значенням від 3 до 20). Знайти модуль вектора.
18	Визначити суму максимального елемента та його індекса. Вивести на екран всі отримані дані.
19	Знайти кількість елементів, що є парними.
20	Визначити, яких елементів більше: парних чи непарних, і наскільки більше.
21	Розташувати елементи масива у порядку спадання. Обчислити суму максимального і мінімального елементів масива.
22	Обчислити суму додатніх значень масиву та мінімальний від'ємний елемент
23	Чи належить масиву задане число (яке користувач вводить з клавіатури)? Якщо належить, то на якій позиції знаходиться?(У випадку, якщо число зустрічається у масиві декілька раз, то у відповіді записати номер однієї з позицій)
24	Розташувати елементи з 1 по 10 за зростанням, а з 11 по 20 за спаданням.
25	Визначити кількість чисел, що знаходяться між максимальним і мінімальним елементами масива.
26	Визначити кількість від'ємних, добуток додатніх і кількість нульових елементів масива.
27	Обчислити суму і кількість парних додатніх елементів.
28	Замінити елементи, що кратні трьом на суму непарних за значенням елементів
29	Визначити суму елементів з парним індексом та добутком елементів непарних за значенням
30	Визначити, яких елементів масиву більше: парних або непарних. Вивести на екран їхню кількість.
31	Обчислити середнє арифметичне парних елементів масиву. Вивести на екран отримані результати
32	Вивести на екран елементи масиву, які більші за 10. Порахувати та вивести на екран їхню кількість
33	Визначити найбільший та найменший елементи масиву та їх порядкові номери. Отримані дані вивести на екран у вигляді текстового повідомлення.
34	Обчислити середнє геометричне додатніх елементів масиву. Вивести на екран кількість непарних елементів масиву
35	Обчислити суму елементів масиву, які кратні 5. Вивести на екран отримані дані та кількість таких елементів масиву
36	Знайти найменший елемент масиву та поміняти його місцями з останнім елементом. Вивести на екран модифікований масив.
37	Замінити в масиві всі додатні елементи одиницями, всі від'ємні – нулями. Вивести на екран модифікований масив
38	Обчислити суму квадратів перших p (вводить користувач з клавіатури) елементів масиву.
39	Обчислити суму квадратів всіх елементів масиву більших за a (вводить користувач з клавіатури)
40	Обчислити кількість від'ємних та нульових елементів масиву.

Завдання 2. Двовимірні масиви. Задачі всіх варіантів передбачають генерування двовимірного масиву випадкових чисел у діапазоні $[-20;20]$. Розмір масиву 10×10 (орієнтовно, для тестування використовуйте масив меншої розмірності). Передбачити в програмі можливість введення з клавіатури розмірності масиву та діапазону для генерування елементів масиву. Обов'язково реалізувати детальний вивід на екран (друк початкової матриці та матриці отриманої після перетворення).

Використовувати додаткову матрицю для проміжних перетворень заборонено!

№ вар.	Завдання
1	Знайти максимальний елемент матриці та обміняти місцями рядок та стовпець до яких він належить.
2	Транспонувати матрицю та обчислити суму всіх її елементів.
3	Обчислити кількість локальних максимумів матриці. Локальний максимум – елемент, який є більшим за всі сусідні елементи.
4	Обміняти місцями рядок, в якому знаходиться мінімальний додатний елемент матриці, та стовпець, в якому знаходиться максимальний від'ємний.
5	Дано двовимірний масив. Поміняйте місцями перший рядок із рядком у якому міститься перший зустрічений мінімальний елемент масиву
6	Обчислити суму модулів кожного рядка, та обміняти місцями рядки з максимальною та мінімально сумами.
7	Дано цілочисельну матрицю. Замінити в ній всі від'ємні елементи, сума індексів яких є парною, на число 95.
8	Знайти кількість елементів масиву, що менші за введене число з клавіатури. Обчислити і вивести на екран суму, добуток, середнє арифметичне чисел з головної діагоналі.
9	Знайти в кожному рядку найменший елемент і поміняти його місцями з відповідним елементом головної діагоналі
10	«Відзеркалити» елементи матриці відносно горизонтальної вісі. Знайти максимальний елемент з «периметру».
11	Знайти середнє значення елементів «периметру». Поміняти місцями елементи головної та побічної діагоналей.
12	Обчислити кількість локальних мінімумів матриці. Локальний мінімум – елемент, який є меншим за всі сусідні елементи.
13	«Віддзеркалити» елементи матриці відносно вертикальної вісі. Знайти «адресу» мінімального за модулем елемента.
14	Обчислити добуток модулів кожного стовпця, та обміняти місцями стовпці з максимальним та мінімальним добутками.
15	Обчислити суму елементів з непарною сумою індексів.
16	Знайти та вивести місцезнаходження мінімального елемента серед максимумів рядків.
17	Обміняти місцями стовпець, в якому знаходиться максимальний від'ємний елемент матриці, та рядок, в якому знаходиться мінімальний додатний.
18	Написати програму, яка в матриці цілих чисел знаходить всі елементи, абсолютні значення яких перевищують задане число В (яке користувач вводить з клавіатури). Обчислити суму та кількість таких елементів.
19	Знайти максимальний елемент квадратної матриці цілих чисел серед елементів, які знаходяться нижче бічної діагоналі.

№ вар.	Завдання
20	Визначити, чи квадратна матриця симетрична відносно головної діагоналі
21	Всі парні елементи квадратної матриці поділити на мінімальний елемент матриці. Вивести на екран мінімальний елемент та матрицю отриману після перетворення.
22	Знайти мінімальний елемент матриці (або один з них). Записати нулі в рядок і стовпчик, на перетині яких він знаходиться.
23	Відсортувати рядок p (номер рядка вводить користувач з клавіатури) квадратної матриці за спаданням. Вивести на екран кількість перестановок та матрицю отриману після перетворення.
24	Знайти максимальний елемент квадратної матриці серед елементів, які знаходяться вище головної діагоналі.
25	Написати програму, яка в матриці A знаходить елементи, що дорівнюють числу введеному з клавіатури. Обчислити кількість таких елементів
26	Дано матрицю A . Замінити в ній всі від'ємні елементи першого рядка на число 0.
27	Визначити, чи є симетричним масив відносно головної діагоналі.
28	Вивести на екран номер рядка, який має максимальну кількість однакових елементів
29	Знайти в кожному рядку найбільший елемент і поміняти його місцями з відповідним елементом головної діагоналі
30	Збільшити кожний додатній елемент масиву в 3 рази. Новий масив виведіть на екран
31	Вкажіть стовпчик, в якому відсутні елементи кратні 9.
32	Поміняйте місцями найбільший парний і найменший непарний елементи у двох стовпцях, номери яких введе користувач з клавіатури
33	Визначити, скільки раз зустрічається число введене з клавіатури серед елементів масива
34	«Відзеркалити» елементи матриці відносно вертикальної вісі. Знайти «адресу» максимального за модулем елемента
35	Обчислити кількість локальних мінімумів матриці. Локальний мінімум – елемент, який є меншим за всі сусідні елементи.
36	Знайти мінімальний елемент квадратної матриці цілих чисел серед елементів, які знаходяться нижче головної діагоналі
37	Відсортувати за зростанням елементи рядка p квадратної матриці цілих чисел. Вивести на екран модифіковану матрицю
38	Визначити кількість елементів квадратної матриці, які дорівнюють p (вводить користувач з клавіатури)
39	В прямокутній матриці знайти рядок з максимальною сумою елементів. Вивести на екран відповідне повідомлення
40	Знайти максимальні елементи в кожному стовпчику прямокутної матриці.

Завдання 3. Двовимірні масиви. Завдання всіх варіантів передбачають генерування двовимірного масиву випадкових чисел у діапазоні $[-20;20]$. Розмір масиву 10×10 . В завданні необхідно надрукувати ті елементи масиву і в такому вигляді, який відповідає кольоровій області на схематичному зображенні двовимірного масиву. Також вивести кількість елементів, які містяться у зафарбованій області та обчислити середнє арифметичне цих елементів масиву. Передбачити в програмі можливість введення з клавіатури розмірності масиву та діапазону для генерування елементів масиву.

Для реалізації завдання необхідно скласти складну умову (із застосуванням логічних операторів) для індексів елементів масиву (див. [довідкові матеріали](#)).

Орієнтовно, відповідь до завдання виглядатиме так:

<pre> Enter interval for array generation -20 20 Enter size of array 10 4 6 -2 7 -6 19 -5 18 -5 8 -3 -1 -17 -20 5 12 5 12 19 11 -10 2 20 18 15 -12 1 -10 12 8 -13 -20 15 -12 10 9 17 -13 12 17 8 0 -8 10 7 -7 -8 20 -1 7 18 8 -2 -8 -13 -15 8 5 -19 -1 17 -11 13 -15 6 -15 5 1 9 8 2 -5 13 -6 -13 19 -19 -15 17 17 1 -8 9 -5 -12 5 16 -9 -4 17 -8 -13 -7 18 -1 12 -13 5 -9 7 start printing -3 -10 2 -13 -20 15 8 0 -8 10 18 8 -2 -8 -13 -15 8 5 -19 -1 5 1 9 8 -15 17 17 -4 17 7 number of elements in picture 30 average of picture elements 0.800000 Process returned 0 (0x0) execution time : 11.096 s Press any key to continue. </pre>	<pre> Enter size of array 10 -20 -14 0 -9 15 -13 -8 -12 4 -9 -7 9 -19 5 10 11 -16 3 6 -1 11 -6 8 -18 14 -11 8 -5 -16 3 8 -17 -20 11 17 -12 -1 -13 8 -20 -5 20 12 -14 -5 -18 -5 -14 -9 -18 -5 -9 -20 -2 -12 14 -6 7 -10 19 -1 5 -4 2 5 -8 -14 -11 -5 15 -19 0 1 6 -7 14 -5 4 18 10 11 12 11 13 11 17 11 7 9 18 -19 -20 4 0 18 -9 -9 4 15 -5 start printing -13 -8 -12 4 -9 11 -16 3 6 -11 8 -5 -12 -1 -18 -12 14 2 5 -8 -14 1 6 -7 14 -5 4 12 11 13 11 17 11 7 9 -19 -20 4 0 18 -9 -9 4 15 -5 number of elements in picture 45 average of picture elements -0.066667 Process returned 0 (0x0) execution time : 16.210 s Press any key to continue. </pre>
--	---

Рисунок 4.8 – Зображення картинок з варіанта №7 та №13 відповідно.

Варіант	Схема	Варіант	Схема	Варіант	Схема
1		2		3	
4		5		6	
7		8		9	
10		11		12	

Варіант	Схема	Варіант	Схема	Варіант	Схема
13		14		15	
16		17		18	
19		20		21	
22		23		24	
25		26		27	
28		29		30	
31		32		33	
34		35		36	

Варіант	Схема	Варіант	Схема	Варіант	Схема
37		38		39	
40		41		42	

Довідкові матеріали до виконання завдання №3

Розглянемо масив, в якому співпадають кількість рядків і стовпців (квадратна матриця). Позначення: i – індекс рядка, j – індекс стовпчика, N – кількість елементів в рядку (стовпці). Далі наведено декілька картинок (кожна з яких схематично зображує двовимірний масив), в яких зафарбована область (тобто елементи, які знаходяться на обраних позиціях) відповідає нерівності, якій задовольняють індекси елементів масива:

$j < N/2$ 	$i < N/2$ 	$j > N/2$ 	$i > N/2$
Індекси елементів, які розташовано на головній діагоналі $i = j$ 	Індекси елементів, які розташовано на побічній діагоналі $i = N - j - 1$ 	Елементи розташовані нижче головної діагоналі $i > j$ 	Елементи розташовані вище головної діагоналі $i < j$
$i > N - j - 1$ 	$i < N - j - 1$ 		

Контрольні запитання

1. Якими є властивості масивів даних?
2. Як здійснюється доступ до елементів одновимірного масиву та матриці?
3. Яким чином можна згенерувати послідовність випадкових чисел на деякому інтервалі?
4. Поясніть значення `srand(time(NULL))` ?
5. Чи можна, не використовуючи циклів, поміняти місцями рядки (стовпці) матриці?
6. Що таке послідовний пошук з бар'єром?

Корисні посилання

1. Оператор `sizeof`. URL: <https://learn.microsoft.com/en-us/cpp/c-language/sizeof-operator-c?view=msvc-170#syntax>
2. Генерування рандомних чисел. URL: <https://acode.com.ua/urok-74-generatsiya-randomnyh-chysel/>
3. Масиви. URL: <https://acode.com.ua/urok-77-masyvy/>
4. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
5. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html>

Лабораторне заняття № 5. Показчики. Створення динамічного масиву

Мета: ознайомитись з поняттям показчика та набуті практичних навичок опрацювання показчиків та динамічних масивів у програмах.

План

1. Основні поняття. Показчики у C/C++.
2. Показчики на показчики.
3. Динамічні масиви в C/C++.
4. Завдання до лабораторної роботи.
5. Контрольні запитання.
6. Корисні посилання.

5.1 Основні поняття. Показчики у C/C++

Показчик – змінна, значенням якої є адреса комірки пам'яті. Показчик може посилатися на змінну або функцію. Для цього потрібно знати адресу змінної або функції. Щоб дізнатися адресу конкретної змінної в C/C++ існує унарна операція взяття адреси **&**.

Щоб отримати значення, записане в деякій області, на яке посилається показчик потрібно скористатися операцією **розіменування** показчика *****. Необхідно поставити зірочку перед ім'ям і отримаємо доступ до значення показчика. Розглянемо програму, яка використовуватиме показчики (див. рис. 5.1).

```
#include <stdio.h>

int main() {
    int var = 123; // ініціалізація змінної var числом 123
    //присвоїли адресу змінної var показчику ptrvar
    int* ptrvar = &var;
    //адресу змінної var отримано операцією взяття адреси
    printf(" &var      = %p\n", &var);
    //адреса змінної var є значенням показчика ptrvar
    printf(" ptrvar    = %p\n", ptrvar);
    //значення змінної var
    printf(" var       = %d\n", var);
    //значення var отримано через розіменування показчика ptrvar
    printf(" *ptrvar   = %d\n", *ptrvar);
    return 0; }
```

Рисунок 5.1 – Лістинг програми з показчиками

Далі наведено результат роботи програми наведеної на рисунку 5.1

```
&var      = 004FF810
ptrvar     = 004FF810
var        = 123
*ptrvar    = 123
```

Рисунок 5.2 – Результат роботи програми

Як видно з результату роботи програми, доступ до значення змінної **var** отримано через безпосереднє звернення до неї та через розіменування покажчика **ptrvar**

З арифметичних операцій, найчастіше використовуються операції додавання, віднімання, інкремент і декремент, оскільки за допомогою цих операцій, наприклад в масивах, обчислюється адреса наступного елементу.

5.2 Покажчики на покажчики

Покажчики можуть посилатися на інші покажчики. При цьому в комірках пам'яті, на які посилатимуться перші покажчики, будуть міститися не значення, а адреси інших покажчиків. Число символів ***** при оголошенні покажчика показує порядок покажчика. Щоб отримати доступ до значення, на яке посилається покажчик його необхідно розіменовувати відповідну кількість разів. Розглянемо програму, яка буде виконує деякі операції з покажчиками порядки вище першого (див. рис. 5.3):

```
int main()
{
    int var = 123; // ініціалізація змінної var числом 123
    int* ptrvar = &var; //покажчик на змінну var
    int** ptr_ptrvar = &ptrvar; // покажчик на покажчик на var
    int*** ptr_ptr_ptrvar = &ptr_ptrvar;
    printf("    var                = %d\n", var);
    printf("    *ptrvar            = %d\n", *ptrvar);
    // двічі розіменовуємо покажчик, оскільки він другого порядку
    printf("    **ptr_ptrvar = %d\n", **ptr_ptrvar);
    // покажчик третього порядку
    printf("    ***ptr_ptr_ptrvar = %d\n", ***ptr_ptr_ptrvar);
    printf("\n***ptr_ptr_ptrvar -> \t***ptr_ptr_ptrvar -> \t*ptrvar
-> \t\tvar -> %d\n", var);
    printf("%p -> %p -> %p -> %p -> %d", &ptr_ptr_ptrvar,
    &ptr_ptrvar, &ptrvar, &var, var);
    printf("\n");
    return 0;}
```

Рисунок 5.3 – Покажчики різних порядків

5.3 Динамічні масиви в C/C++

В C/C++ операції **new** і **delete** призначені для динамічного розподілу пам'яті комп'ютера. Операція **new** виділяє пам'ять з області вільної пам'яті, а операція **delete** звільняє виділену пам'ять. Пам'ять, яка виділяється, після її використання повинна вивільнятися, тому операції **new** і **delete** використовуються парами

Найчастіше операції **new** і **delete** застосовуються для створення динамічних масивів, а не для створення динамічних змінних. Розглянемо фрагмент коду створення одновимірної динамічного масиву (див. рис. 5.4).

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
    srand(time(0));
    float* ptrarray = new float[10]; //створення масиву з 10
    дійсних чисел
    for (int count = 0; count < 10; count++) {
        // заповнення масиву випадковими числами від 1 до 10
        //звернення до елементів масиву через покажчик
        ptrarray[count] = (rand() % 10 + 1) / float((rand() % 10 +
1));
    }
    for (int count = 0; count < 10; count++) {
        printf(" ptrarray[%d] = %.2f, %p \n", count,
ptrarray[count]);
    }
    delete[] ptrarray; // вивільнення пам'яті
    return 0;}

```

Рисунок 5.4 – Лістинг програми генерування динамічного масиву

Для друкування адрес елементів масиву на екран необхідно застосувати специфікатор формату **%p**, а в якості аргумента до функції **printf()** передати **&ptrarray[count]**.

При передаванні масивів до функцій, треба пам'ятати:

1. До функції масиви завжди передаються через адресу початку масиву;
2. Ім'я масиву являє собою константний покажчик на початок (на нульовий елемент) масиву, тобто наступні записи є еквівалентними:

mas == &mas == &mas[0]

Тепер розглянемо фрагмент коду, в якому показано, як оголошується двовимірний динамічний масив. Оголошення та видалення двовимірного динамічного масиву виконується за допомогою циклу, тому розглянемо програму, в якій створюється двовимірний динамічний масив (див. рис. 5.5).

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
    srand(time(0));
    int row = 4, column = 5;
    float** ptrarray = new float* [row]; // створення масиву
    покажчиків на рядки
    for (int i = 0; i < row; i++)

```

```

        ptrarray[i] = new float[column]; // створення масиву
стовпців
    for (int i = 0; i < row; i++)
        for (int j = 0; j < column; j++)
            ptrarray[i][j] = (rand() % 10 + 1) / float((rand() % 10 + 1));
    printf(«\n»);
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < column; j++)
            printf(« %.2f », ptrarray[i][j]);
        printf(«\n»);
    }
    // вивільнення пам'яті
    for (int i = 0; i < row; i++)
        delete[] ptrarray[i];
    return 0;}

```

Рисунок 5.5 – Генерування двовимірного динамічного масиву

Спочатку оголошується покажчик другого порядку **float **ptrarray**, який посилається на масив покажчиків **float* [row]**, де розмір масиву дорівнює чотирьом. Після чого в циклі **for** кожному рядку масиву оголошеного в рядку №2 виділяється пам'ять під п'ять елементів. В результаті отримуємо двовимірний динамічний масив **ptrarray[4][5]**.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №5

Відповідно до варіанта (визначається за списком академгрупи) виконати 2 завдання. В кожній програмі реалізувати нескінченний цикл. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1, 2;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта
3. Умови завдань 1-2, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №5

Завдання 1.

- Написати програму для виконання певних дій над **динамічним одновимірним масивом**, реалізувавши доступ до елементів масива двома способами: за допомогою індексів та покажчиків.

- Алгоритм формування елементів масива реалізувати двома способами: за допомогою введення даних з клавіатури та генерації випадкових чисел. Передбачити виділення пам'яті під динамічний масив функціями **malloc**, **new**.

- Вивести на екран елементи початкового масива та відповідні адреси цих елементів. Якщо масив було змінено, то у новому рядку вивести модифікований масив.

- Введення-виведення даних супроводжувати відповідними повідомленнями.

- В програмі реалізувати нескінченний цикл.

- Обробити створений масив відповідно до варіанта завдання.

№	Завдання
1	Підрахувати кількість елементів масиву E, рівних нулю, індекси яких парні.
2	Обчислити добуток ненульових елементів масиву F, індекси яких непарні.
3	Обчислити суму елементів масиву M, значення яких перевищує індекс елемента
4	Обчислити суму елементів масиву B, значення яких кратне п'яти і не перевищує задане число A.
5	Обчислити добуток елементів масиву S, значення яких не більше заданого
6	Знайти максимальне число серед елементів масиву U, значення яких непарне.
7	Знайти номери елементів масиву S, значення яких парне.
8	Обчислити суму останніх десяти елементів масиву B.
9	Обчислити суму елементів масиву H, що не належать інтервалу [A, B].
10	Підрахувати кількість не додатних, додатних, нульових елементів масиву D.
11	Знайти середнє арифметичне елементів Q, значення яких менше A.

№	Завдання
12	Знайти середнє геометричне елементів R, значення яких більше A.
13	Обчислити суму творів значень елементів масиву G на значення їх індексів.
14	Обчислити суму індексів елементів масиву W, у яких значення елементів більше A.
15	Обчислити суму всіх елементів масиву V, при обчисленні не додатні значення елементів замінювати їх абсолютним значенням.
16	Обчислити квадрат суми елементів, які більше заданого числа A і індекси яких кратні 3.
17	Вивести на друк елементи масиву Z, квадрат значення яких перевищує значення індексу в кубі.
18	Порівняти суму і добуток парних за індексом елементів масиву B, виключаючи нульові елементи.
19	Підрахувати кількість додатних елементів масиву X, значення яких в три рази більше індексу.
20	Обчислити добуток елементів масиву Y, значення яких не перевищує $3*i$, де i – індекс елемента.
21	Обчислити суму елементів масиву C, значення яких парне і не перевищує задане число A.
22	Обчислити суму квадратів елементів масиву B, менших заданого числа A
23	Обчислити добуток елементів масиву S, значення яких не менше заданого позитивного числа A, а індекси кратні трьом.
24	Обчислити суму парних елементів масиву F, що перевищують задане число A.
25	Підрахувати кількість не додатних елементів масиву B, значення яких по модулю перевищує індекс елемента.
26	Знайти мінімальне число серед елементів масиву E, індекс яких кратний трьом.
27	Знайти номери елементів масиву D, значення яких непарне.
28	Обчислити добуток перших семи елементів масиву A.
29	Знайти номери елементів масиву M, рівних заданому числу A
30	Обчислити добуток елементів масиву C, що належать інтервалу [A,B].
31	Написати програму для визначення суми елементів з парними ідексами
32	Визначити значення та порядкові номери двох найбільших елементів вектора
33	У векторі дійсних чисел визначити значення та порядковий номер максимального від'ємного елемента цього вектора
34	Обчислити кількість непарних елементів масива, кратних 3
35	Обчислити кількість невід'ємних елементів масива
36	У векторі дійсних чисел визначити значення та порядковий номер мінімального додатного елемента цього вектора
37	Сформувати вектор з 10 непарних чотирицифрових чисел без повторень елементів. Визначити мінімальний і максимальний елементи вектора та поміняти їх місцями.
38	Обчислити середнє арифметичне додатних елементів масива з непарними індексами. Вивести ці елементи на екран
39	Знайти мінімальний елемент масиву серед елементів другої половини масиву
40	Відсортувати масив за спаданням його елементів. Додатково вивести на екран кількість перестановок

Завдання 2.

- Написати програму для виконання певних дій над **динамічним двовимірним масивом**, реалізувавши доступ до елементів масива двома способами: за допомогою індексів та покажчиків.

- Алгоритм формування елементів масива реалізувати двома способами: за допомогою введення даних з клавіатури та генерації випадкових чисел. Передбачити виділення пам'яті під динамічний масив функціями **malloc**, **new**.

- Вивести на екран елементи початкового масива у вигляді матриці. Якщо масив було змінено, то з нового рядка вивести модифікований масив.

- Введення-виведення даних супроводжувати відповідними повідомленнями.

- В програмі реалізувати нескінченний цикл.

- Обробити створений масив відповідно до варіанта завдання.

№	Завдання
1	В заданій матриці $A(N,M)$ знайти рядки, що не містять від'ємних елементів.
2	Із заданої матриці $A(N,M)$ видалити рядок і стовпець, в яких знаходиться перший елемент, рівний нулю. Отриману матрицю ущільнити. Елементи матриць проглядати зліва направо і зверху вниз.
3	Отримати нову матрицю $S(N, M)$ шляхом складання всіх елементів заданої матриці $C(N, M)$ з її найбільшим за модулем елементом.
4	Замінити мінімальний елемент головної діагоналі матриці $D(N, N)$ сумою елементів першого рядка
5	Знайти і надрукувати, скільки не додатних елементів містяться в кожному рядку матриці $F(N, M)$.
6	Замінити останній стовпець матриці $H(N, N)$ мінімальним елементом головної діагоналі.
7	Розташувати елементи всієї матриці $C(N, M)$ за спаданням (зверху вниз, зліва направо).
8	Обчислити суму рядків матриці $Z(N, N)$ з нульовими елементами на побічній діагоналі.
9	Збільшити значення кожного елемента стовпця матриці $G(N, M)$ на значення першого елемента відповідного стовпця.
10	Знайти і надрукувати значення і індекси мінімального елемента в кожному рядку матриці $W(N, M)$.
11	Обчислити суму парних елементів в кожному стовпці матриці $A(N, M)$
12	Розташувати елементи кожного рядка матриці $A(N, M)$ за спаданням.
13	Отримати нову матрицю $F(N, M)$ шляхом складання всіх елементів заданої матриці $B(N, M)$ з її найменшим за модулем елементом.
14	Поміняти місцями рядок матриці $P(N, N)$, що містить максимальний елемент головної діагоналі, із стовпцем, що містить мінімальний елемент головної діагоналі.
15	Визначити, скільки рядків заданої матриці $U(N,M)$ містять хоча б один елемент із діапазону $[A, B]$.
16	Із заданої матриці $H(N,M)$ видалити рядок, в якому знаходиться перший не додатний елемент. Елементи матриць проглядати зліва направо і зверху вниз
17	В заданій матриці $G(N,M)$ знайти індекси першого елемента, що перевершує середнє арифметичне всіх елементів. Елементи матриць проглядати зліва направо і зверху вниз.
18	В заданій матриці $T(N,M)$, яка містить тільки цілі числа, замінити перший не додатний елемент максимальним елементом матриці. Якщо не додатних елементів немає, то вивести відповідний текст. Елементи матриць проглядати зліва направо і зверху вниз.
19	Отримати нову матрицю $C(N, M)$ шляхом віднімання всіх її елементів з максимального елемента матриці $A(N, M)$.
20	Замінити всі елементи матриці $Y(N, N)$, розташовані вище головної діагоналі, максимальним елементом, розташованим на головній діагоналі.
21	Знайти кількість додатних елементів в кожному стовпці матриці $Z(N, M)$.

№	Завдання
22	Замінити перший рядок матриці $X(N, N)$ максимальним елементом головної діагоналі.
23	Обчислити число непарних елементів в кожному рядку матриці $W(N, M)$
24	Дана матриця $A(N, M)$. Визначити кількість рядків матриці A , що містять хоча б одну нульову компоненту.
25	Обчислити суму стовпців матриці $F(N, N)$ з позитивними елементами на головній діагоналі.
26	Розташувати елементи всієї матриці $B(N, M)$ за збільшенням (зліва направо, зверху вниз).
27	Помножити значення кожного елемента рядка матриці $D(N, M)$ на значення останнього елемента відповідного рядка.
28	Знайти і надрукувати значення і індекси максимального елемента в кожному стовпці матриці $C(N, M)$.
29	Поміняти місцями рядок, що містить максимальний елемент, з рядком, що містить мінімальний елемент матриці $E(N, M)$.
30	Розташувати елементи кожного стовпця матриці $A(N, M)$ в порядку зростання.
31	Визначити суму елементів прямокутної матриці в тих стовпчиках, які не містять нульових елементів.
32	Визначити кількість рядків прямокутної матриці, що не містять жодного нульового елемента, а також максимальне з чисел, що зустрічаються в заданій матриці більше одного разу
33	Визначити добуток елементів прямокутної матриці в тих рядках, які не містять від'ємних елементів, а також максимум серед сум елементів головної і побічної діагоналей
34	У квадратній матриці цілих чисел обчислити суму елементів тих рядків, які не містять від'ємних елементів, а також мінімум серед сум головної та побічної діагоналей
35	Ввести прямокутну матрицю дійсних чисел. Поміняти місцями два рядки матриці, з відповідно найбільшою і найменшою сумою елементів
36	Ввести прямокутну матрицю цілих чисел. Визначити кількості повторень значень елементів матриці
37	Ввести прямокутну матрицю цілих чисел. Відсортувати кожен рядок за зростанням значень елементів. Окремо вивести на екран кількість перестановок
38	Ввести матрицю цілих чисел. Визначити, чи ця матриця є симетричною відносно побічної діагоналі
39	Ввести квадратну матрицю цілих чисел. Знайти мінімальний елемент серед елементів, які знаходяться нижче головної діагоналі
40	Ввести квадратну матрицю дійсних чисел. Якщо сума елементів головної діагоналі більша від суми елементів бокової діагоналі, то виконати транспонування матриці – заміну рядків на відповідні стовпчики

Контрольні запитання

1. Чи є показник змінною? Якого типу?
2. Які арифметичні операції можна здійснити з показником типу **void**?
3. Яка кількість байт виділяється для показника?
4. Як отримати доступ до даних, на які вказує показник?
5. Чим відрізняються такі операції з показником **a**: **(*a+1)** та **(a+1)**?
6. Що таке показники на показники і для чого їх використовують?
7. Що означає поняття «дикий» показник?

Корисні посилання

1. Показчики. URL: <https://acode.com.ua/urok-84-vkazivnyky/#toc-7>
2. Malloc. URL:
<https://pubs.opengroup.org/onlinepubs/009695399/functions/malloc.html>
3. Відлагодження програм: степпінг і точки зупину. URL:
<https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
4. C++ Programming Style Guidelines. URL:
<https://geosoft.no/development/cppstyle.html>

Лабораторне заняття № 6. Символьний тип даних. Рядки

Мета: ознайомитись з представленням рядків в С та набути практичних навичок опрацювання рядків та символьних масивів у програмах.

План

1. Представлення символів в С.
2. Представлення рядків в С.
3. Функції для обробки рядків.
4. Завдання до лабораторної роботи.
5. Контрольні запитання.
6. Корисні посилання.

6.1 Представлення символів в С

Множина допустимих значень символьного (літерного типу) – це множина цілих чисел в діапазоні від 0 до 255. Зберігання значення такого типу потребує одного байта оперативної па'м'яті. Найчастіше такий тип застосовується для зберігання символів кодової таблиці **ASCII**, а отже, даними цього типу є окремі символи. Кожному символу відповідає ціле число (код) в діапазоні від 0 до 127. Як вже говорилося раніше, символьний тип позначається ідентифікатором **char**.

Значення символьного типу записуються в одинарних лапках. Наприклад, **'A'**, **'1'**, **'+'** тощо. На рисунку 7.1 наведено фрагмент програми, який демонструє зв'язок між типами **int** та **char**.

```
char symbol = 'd';
int a = symbol; //це символ 'd' масиву symbol
printf("%d\n", a); //буде надруковано 100, оскільки в ASCII-
таблиці код символу 'd'дорівнює 100
printf("%c\n", symbol); //буде надруковано d
```

Рисунок 6.1 – Зв'язок між типами даних **int** та **char**

Далі, в таблиці 6.1 наведено фрагмент **ASCII**-таблиці з відповідностями між певними символами та їх кодами

Таблиця 6.1 – Відповідність між символом та його кодом

Символ	Код
' ' (пробіл)	32
'0' ... '9'	48 ... 57
'A' ... 'Z'	65 ... 90
'a' ... 'z'	97 ... 122

Перелік деяких керуючих послідовностей, які називають ESC-послідовностями, подано в таблиці 6.2.

Таблиця 6.2 – Деякі керуючі символи таблиці **ASCII** кодів

Символ	Опис
\n	Символ переведення курсору на початок наступного рядку
\r	Перехід на початок поточного рядка (переведення каретки)
\t	Горизонтальна табуляція (відповідає клавіші <Tab>)
\v	Вертикальна табуляція
\b	Символ видалення попереднього символу перед курсором (відповідає клавіші <BackSpace>)
\a	Символ звукового сигналу системного динаміка
\\	Зворотна коса риска
\?	Знак питання
\`	Апостроф (єдинична лапка)
\"	Подвійні лапки
\0	Завершальний символ рядка символів (нуль-символ)
\0oo	Вісімковий код символу, де oo – одна, дві чи три вісімкові цифри (0...7)
\0xhh	Шістнадцятковий код символу, де hh – одна, дві чи більше шістнадцяткових цифр (0...9, A, ..., F)

Для даних символного типу означено операції порівняння. Символи вважають рівними, якщо рівні їх **ASCII**-коди. Один символ вважають більшим за інший, якщо його **ASCII**-код більший. Крім операцій порівняння для даних символного типу означено операції додавання та віднімання, у результаті виконання яких утворюються нові символи. Ці операції позначають символами «+» та «-».

Приклад 6.1 Отримати символ десяткової цифри зі значення цілої змінної, що належить діапазону 0...9.

Отримати такий символ можна таким чином:

```
int n; char c; c = n + '0';
```

Код символу '0' дорівнює 48, а тому додаючи до числа **n** (приймає значення від 0 до 9) число 48 отримуємо число від 48 до 57. А оскільки тип змінної **c** (результат такого додавання) – **char**, то отримане нами число це **ASCII**-код символів '0'...'9'. А обернене перетворення можна зробити за допомогою такої комбінації: `if (c >= '0' && c <= '9') n = c - 0;`

Приклад 6.2 Перетворити маленьку латинську літеру на велику.

Зробити це можна таким чином:

```
if (c >= 'a' && c <= 'z') c = c - 'a' + 'A';
```

В даному фрагменті програми перш за все перевіряється умова, чи є значення змінної маленькою латинською літерою. Потім отримуємо десяткове число (від 0 до 9), а далі додаємо код символу 'A' і таким чином отримуємо велику літеру відповідну до маленької літери.

6.2 Представлення рядків в С

Рядковий тип даних в мові програмування С реалізовано як лінійний масив символів. Для полегшення роботи з рядковими даними в С реалізовано ряд функцій. Кожний символ рядка зберігається в масиві типу **char**. При цьому, останній символ повинен бути `'\0'`.

Далі на рисунку 6.2 наведено декілька прикладів оголошення рядків

```
#include <stdio.h>

int main()
{
    char name1[15];
    char name2[15] = "Coding C";
    char name3[] = { 'a', 'b', 'c', 'd', 'e' };
    char name4[15] = { 'C', 'o', 'd', 'i', 'n', 'g', ' ', ' ', 'C' };
    char S[256] = "";
    printf("%s\n", name1);
    printf("%s\n", name2);
    printf("%s\n", name3);
    printf("%s\n", name4);
}
```

Рисунок 6.2 – Приклад оголошення рядків в програмі

Наведемо в таблиці 6.3 опис декількох функцій для обробки символів:

Таблиця 6.3 – Функції для обробки символів

Прототип функції	Опис
int isalnum (int c);	Перевірка, чи є символ літерою або цифрою
int isalpha (int c);	Перевірка, чи є символ літерою
int iscntrl (int c);	Перевірка, чи є символ керуючим
int isdigit (int c);	Перевірка, чи є символ десятковою цифрою
int isgraph (int c);	Перевірка, чи є символ видимим
int isprint (int c);	Перевірка, чи є символ видимим, включаючи пробіл
int islower (int c);	Перевірка, чи є символ літерою нижнього регістру
int ispunct (int c);	Перевірка, чи є символ знаком пунктуації
int isspace (int c);	Перевірка, чи є символ пробільним
int isupper (int c);	Перевірка, чи є символ літерою верхнього регістру
int isxdigit (int c);	Перевірка, чи є символ шістнадцятковою цифрою
int tolower (int c);	Перетворення символу в нижній регістр
int toupper (int c);	Перетворення символу у верхній регістр

Вищезазначені функції перевірки й перетворення регістру працюють лише з латинськими літерами. Спроба використати ці функції для символів кирилиці спричинить виведення повідомлення про помилку. Тому для роботи з літерами кирилиці слід використовувати перевірку належності символу до

відповідного символічного проміжку, а для перетворення регістру – зменшення чи то збільшення коду символу на різницю кодів між великими та малими літерами (для більшості літер вона становить 32).

6.3 Функції обробки рядків

Для введення одного символу в консолі існує функція **getch** з бібліотеки **conio.h**. Для введення і виведення рядків в консолі використовують такі функції **gets/puts** та **scanf/printf** (з бібліотеки **stdio.h**).

Таблиця 6.4 – Функції обробки рядків

Функція	Опис функції
puts(s)	виводить рядок s на екран, замінюючи нуль-символ на Enter.
gets_s(s, buf)	зчитує символи з клавіатури до появи символу переведення рядка Enter і записує їх у рядок s , параметр buf відповідає за максимальну кількість символів, які буде зчитано
printf("%s", S)	виводить рядок S на екран
scanf_s("%s", S, buf)	зчитує символи з клавіатури і записує їх у рядок S , параметр buf відповідає за максимальну кількість символів, які буде зчитано
strlen(char *s);	повертає довжину рядка (без урахування символу закінчення рядка)
strcpy(char *s1, char *s2);	копіює в s1 рядок s2 , повертає s1 , при цьому старе значення s1 втрачається
strncat(char *s1, char *s2, size_t n);	приєднує до рядка s1 перші n символів з рядка s2
strpbrk(char *s1, const char *s2)	знаходить місце першого входження в рядок s1 будь-якого символу рядка s2 і повертає частину рядка s1 , починаючи з цього символу і до кінця
strtok(char *s1, const char *s2);	повертає частину (лексему) рядка s1 від поточної позиції вказівника до розділового символу, зазначеного у рядку s2

Наведемо декілька прикладів роботи з рядками.

Приклад 6.3 Обчислити довжину рядка.

Текст програми наведено на рисунку 6.3. Для обчислення довжини рядка використано функцію **strlen**, яка приймає в якості аргумента рядок **S**.

```
#include <stdio.h>
#include<string.h>

int main()
{
    char S[6] = "Hello"; // 7 байт заповнено літерами, 8-й байт -
    нуль-символ '\0'
    int lenght = strlen(S);
    printf("Number of symbols in string S %d", lenght);
    return 0;
}
```

Рисунок 6.3 – Обчислення кількості символів у рядку

Приклад 6.4 Вивести всі символи та коди таблиці ASCII

На рисунку 6.4 наведено одну з можливих реалізацій. В даній програмі застосовано приведення типів для виведення на друк символа. Оскільки в якості аргумента до функції **printf** передано **(char)i**, то буде виведено символ, **ASCII**-код якого дорівнює **i**.

```
#include <stdio.h>
#include <string.h>

int main()
{
    for (int i = 32; i < 256; i++) {
        if (i % 4 == 0) printf("\n");
        printf("%3i = %c \t", i, (char)i);
    }
    return 0;
}
```

Рисунок 6.4 – Виведення таблиці ASCII на екран

Приклад 6.5 Розташувати символи рядка в порядку зростання їхніх кодів. Обчислити кількість пробілів.

```
#include <stdio.h>
#include <string.h>

int main()
{
    char testString[256] = ""; //оголошення рядка
    int cnt = 0; //змінна для кількості пробілів
    printf("Enter your string ...");
    gets_s(testString); //зчитування рядка
    printf("You input this string ...%s\n", testString);
    int lenght = strlen(testString); //обчислення довжини введеного рядка
}
```

```

for (int i = 0; i < lenght; i++) {
    if (testString[i] == ' ')//або так testString[i]==32
        cnt++;
}
printf("number of spaces is %d", cnt);
//сортування символів рядка за зростанням їх кодів
for (int i = 0; i < lenght; i++) {
    for (int j = 0; j < lenght - i - 1; j++) {
        if (testString[j] > testString[j + 1]) {
            char temp = testString[j];
            testString[j] = testString[j + 1];
            testString[j + 1] = temp;
        }
    }
}
printf("\nString after sorting ... %s\n\n", testString);
}

```

Рисунок 6.5 – Сортування елементів рядка та обчислення кількості пробілів

Приклад 6.6 Користувач задає інвертований рядок (тобто записаний справа наліво), а комп'ютер відображає правильний рядок та рахує кількість слів у ньому. Словом вважається набір символів між двома пробілами. Реалізація даної задачі має вигляд (див. рис. 6.6)

```

#include <stdio.h>
#include <string.h>

int main()
{
    char S[256], newS[256]; //змінні, що містять введений рядок та
    результат
    printf("Input Reverse String...\n");
    gets_s(S); //зчитали рядок з клавіатури
    printf("You input this string: %s \n", S);
    char* next_token = NULL;

    //блок інвертування рядка
    int n = strlen(S); //визначення довжини рядка
    for (int i = 0; i < n; i++)
        newS[i] = S[n - i - 1]; //цикл, що інвертує заданий рядок
    newS[n] = '\0'; //обов'язковий символ закінчення рядка
    printf("The Result String is -%s- \n", newS);
    char *word; //показчик на початок слова
    word = strtok_s(newS, " ", &next_token); //виділяємо слово між
    пробілами
    while (word != NULL) {
        printf("\n\t%s", word);
        word = strtok_s(NULL, " ", &next_token);
    }
    return 0;
}

```

Рисунок 6.6 – Лістинг програми інвертування рядка

Приклад 6.7 Видалити з рядка всі слова, які закінчуються на голосну.
Одна з можливих реалізацій даного завдання наведена на рисунку 6.7.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>

int main()
{
    while (1)
    {
        char S[256], newS[256] = {0}; // оголошуємо два рядки
        char vowels[] = "aeiouy"; // створюємо масив з голосними
        int i, lennewS; // змінна для довжини нового рядка
        char *next_token = NULL;
        printf(" \nInput string: \n");
        gets_s(S);
        printf(" You input this string: \n%s\n", S);
        char* word; // покажчик на адресу пам'яті, де зберігатиметься
        слово word
        word = strtok_s( S , " ", &next_token); // виділяємо слово, тобто
        послідовність символів між пробілами
        while (word != NULL) // поки не досягнуто кінець рядка
        {
            char isvowel = 0; // змінна, яка сигналізуватиме, чи
            закінчується слово на голосну
            int lenw = strlen(word); // довжину вирізаного з рядка слова
            зберігаємо в змінній lenw (знадобиться пізніше)
            for (i = 0; i < sizeof(vowels); i++){
                if (word[lenw - 1] == vowels[i]) // послідовно порівнюємо
                останню букву слова з елементами масива vowels
                {
                    isvowel = 1; // якщо це так, то слово нам не цікаве,
                    робимо присвоєння і break
                    break; // переходимо на рядок 39, а далі на 24
                }
            }
            if (!isvowel) // якщо зустріли слово з приголосною
            наприкінці, то формуємо з них рядок без голосного закінчення
            {printf("Add to result string this word: %s\n ", word);
                strcat_s(newS, word); // об'єднуємо рядок newS і рядок
            word
                strcat_s(newS, " "); // додаємо пробіл, щоб слова не
            поєднувались при друці
            }
            word = strtok_s(NULL, " ", &next_token); // переходимо до
            наступного слова
        }
        lennewS = strlen(newS); // знаходимо довжину нового рядка, щоб на
        правильну позицію поставити символ кінця рядка
        newS[lennewS] = '\0';
        printf("\n New string without vowels in the end of words \n");
        printf("\n %s \n", newS);
        _getch();
    }
    return 0;
}
```

Рисунок 6.7 – Лістинг програми видалення з рядка слів

Декілька коментарів до цієї програми. По-перше, задачу про видалення слів з рядку **S** переформульовано у задачу формування нового рядка **newS**, який складатиметься зі слів без голосної в кінці слова. Фрагмент програми, який відповідає за порівняння останньої букви кожного слова з елементами масиву **vowels** можна замінити відповідною бібліотечною функцією **strrchr()** з бібліотеки **string.h**.

Зауваження! Для коректної роботи програми необхідно підключити наступні модулі: **<stdio.h>**, **<stdlib.h>** та **<string.h>**.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №6

Відповідно до варіанта (визначається за списком академгрупи) виконати 2 завдання. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1, 2;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта.
3. Умови завдань 1-2, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №6

Завдання 1. Напишіть програму відповідно до завдання. Обов'язково використати консольне меню користувача, символьний масив згенерувати за допомогою функції `rand()` та генератора `srand(time(NULL))`. Для контролю коректності введення вхідних даних виконувати ехо-виведення їх значень на екран

Зауваження.

- Використання глобальних змінних не допускається.
- **Забороняється використовувати тип даних `string`.** Використовувати лише масив символів.

№ вар.	Завдання
1	Створіть масив з n (вводить користувач з клавіатури) елементів. Замініть символи-цифри на символ «*»
2	Створіть масив з n (вводить користувач з клавіатури) елементів. Переведіть всі строчні букви-символи у верхній регістр
3	Створіть масив з n (вводить користувач з клавіатури) елементів. Замініть кожен символ-цифру на символ «!»
4	Створіть масив з n (вводить користувач з клавіатури) елементів. Якщо символ є прописною буквою латинського алфавіту (тобто буквою від a до z), то замініть попередній символ на пробіл
5	Створіть масив з n (вводить користувач з клавіатури) елементів. Поміняйте регістр символів: прописні замініть на строчні, строчні - на прописні.
6	Дано матрицю символів розміром $m \times n$ (вводить користувач з клавіатури). Скільки разів серед даних символів зустрічаються символи «+», «-», «*».
7	Створіть масив з n (вводить користувач з клавіатури) елементів. Обчисліть кількість голосних латинських букв
8	Дано масив символів (кількість елементів вводить користувач з клавіатури), серед яких є двокрапка «:». Визначте, скільки символів йде перед ним.
9	Дано масив символів (кількість елементів вводить користувач з клавіатури), який

№ вар.	Завдання
	містить англійський текст. Знайти кількість слів, що починаються з букви <i>b</i> .
10	Дано масив символів (кількість елементів вводить користувач з клавіатури). Скільки в ньому символів <i>r, k, t</i> .
11	Дано масив символів (кількість елементів вводить користувач з клавіатури). Визначте, скільки в ньому символів «*», «;», «:»
12	Дано масив символів (кількість елементів вводить користувач з клавіатури), серед яких є символ «#». Визначте, скільки символів йде за ним.
13	Дано масив символів (кількість елементів вводить користувач з клавіатури). Визначте, скільки разів міститься в ньому група символів <i>abc</i> .
14	Дано масив символів (кількість елементів вводить користувач з клавіатури). Замініть всі двокрапки «:» на крапку з комою «;». Визначте кількість замін.
15	Дано масив символів (кількість елементів вводить користувач з клавіатури). Видаліть з нього символ двокрапка «:», обчисліть кількість видалених символів
16	Дано масив символів (кількість елементів вводить користувач з клавіатури). Видаліть частину тексту, що міститься в дужках (разом з дужками)
17	Дано масив символів (кількість елементів вводить користувач з клавіатури), в якому є один символ \$. Обчисліть кількість символів до цього символу і після нього.
18	Із заданого символічного рядка виберіть ті символи, які містяться в ній тільки один раз, в тому порядку, в якому вони містяться в тексті.
19	Дано масив символів (кількість елементів вводить користувач з клавіатури). Перевірити, чи однакова кількість відкриваючих і закриваючих квадратних дужок в даному рядку.
20	Дано матрицю символів розмірності 3×4. Обчислити кількість символів «@» в цьому масиві.
21	Дано матрицю символів розмірності 5×4. Обчислити кількість символів «+» і «*» в цьому масиві.
22	Згенеруйте масив із 12 символів. Подвійте всі ASCII-коди символів масиву.
23	Дано матрицю символів розмірності 3×4. Обчислити кількість цифр в цьому масиві. Замініть їх знаком «!»
24	Дано масив символів, який містить текст (текстом вважати послідовність різних символів та пробілів). Визначте довжину третього слова (словом вважати послідовність різних символів між пробілами).
25	Дано масив символів. Впорядкувати символи рядка за зростанням їх кодів.
26	Дано матрицю символів розмірності 4×4. Визначте суму кодів символів елементів головної діагоналі.
27	Дано матрицю символів розмірності 5×5. Визначити символ з максимальним кодом
28	Дано масив символів. Замінити в ньому кожну крапку багатокрапкою («...»)
29	Дано масив символів. Вкажіть слова, в яких міститься хоча б одна буква <i>k</i> .
30	Дано матрицю символів розмірності 5×7. Визначте суму цифр в матриці.
31	Дано масив символів, що складається з англійських слів та пробілів. Впорядкуйте даний масив англійських слів за алфавітом. Словом вважати послідовність символів, яка міститься між пробілами.
32	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Замініть символи-цифри на символ «*»
33	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Переведіть всі строчні букви-символи у верхній регістр
34	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Замініть кожен символ-цифру на символ «!»
35	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Якщо символ є

№ вар.	Завдання
	прописною буквою латинського алфавіту (тобто буквою від <i>a</i> до <i>z</i>), то замініть попередній символ на пробіл
36	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Поміняйте регістр символів: прописні замініть на строчні, строчні - на прописні.
37	Дано матрицю символів розміром $m \times n$ (вводить користувач з клавіатури). Скільки разів серед даних символів зустрічаються символи «+», «-», «*».
38	Створіть масив з <i>n</i> (вводить користувач з клавіатури) елементів. Обчисліть кількість приголосних звуків.
39	Дано масив символів, серед яких є двокрапка «:». Визначте, скільки символів міститься перед ним.
40	Дано масив символів, який містить англійський текст (текстом вважати послідовність символів, букв та пробілів). Знайти кількість слів, що починаються з букви <i>b</i> .

Завдання 2. Створити програму згідно свого варіанта. Обов'язково створити консольне меню користувача. Рядок ввести з клавіатури. Для контролю правильності введення вхідних даних виконувати ехо-виведення їх значень на екран.

Примітка. Словом вважати послідовність букв між пробілами. Реченням вважати послідовність букв, пробілів та розділових знаків. За необхідності використовуйте додатковий рядок. Скористайтесь прикладами 6.6 та 6.7

Зауваження.

- Використання глобальних змінних не допускається
- **Забороняється використовувати тип даних `string`.** Використовувати лише масив символів

№ вар.	Завдання
1	В заданому рядку кожен символ «!» замінити символом « » (пробіл). Результат вивести на екран.
2	В заданому рядку знайти розташування першого символу «*».
3	Видалити з рядка всі слова, що починаються з голосної літери.
4	У заданому рядку підрахувати кількість букв А
5	У заданому рядку видалити перший символ
6	У заданому рядку додати в кінець символ «»
7	У заданому рядку видалити всі символи, що знаходяться на парному індексі
8	У заданому рядку після кожної крапки поставити пробіл.
9	Задано рядок. Дописати в кінець рядку його довжину
10	Записати речення латиницею відповідно до правил транслітерації.
11	Підрахувати у рядку кількість слів, що починаються та закінчуються з однієї літери.
12	У рядку, що є алгебраїчним виразом, перевірити коректність розставлених дужок.
13	У рядку підрахувати кількість чисел (але не цифр).
14	У заданому рядку розташувати всі символи за зростанням їх кодів.
15	Видалити всі зайві символи пробілів.
16	У рядку визначити довжину найдовшого слова та видалити його.
17	Поміняти місцями всі слова в заданому реченні (перше слово з останнім, друге з передостаннім тощо).
18	У заданому рядку після кожного символу вставити код, який відповідає цьому

№ вар.	Завдання
	символу.
19	Задано рядок. Отримати (вивести на друк) передостаннє слово цього рядка. Роздільником слів вважати один або декілька пробілів.
20	Визначити, чи входить у заданий рядок пробіл, якщо так – визначити кількість.
21	Ввести речення та вивести на екран ті його слова, які містять подвоєння літер або вивести повідомлення про відсутність таких слів.
22	В заданому рядку після кожної крапки поставити ще одну крапку.
23	В рядку підрахувати кількість голосних та приголосних літер, видалити голосні.
24	Видалити з рядка всі слова, що розташовані за розділовими знаками.
25	В заданому рядку підрахувати кількість ком і крапок.
26	В кінці заданого рядка дописати символи « », довести довжину рядка до 25.
27	Видалити у заданому рядку все, окрім першого і останнього символів.
28	В заданому рядку знайти місцезнаходження першого символу «*».
29	В рядку розташувати всі символи за зростанням їхніх кодів.
30	В рядку після кожного слова дописати його довжину.
31	Ввести речення та обчислити середню довжину слова. Розділові знаки та числа не враховувати.
32	У рядку замінити кожну букву символом «*».
33	Ввести речення та видалити з нього всі слова, які містять задані користувачем букви.
34	Ввести речення і обчислити кількість слів у реченні. Словом вважати послідовність символів між пробілами.
35	Ввести речення та вивести на екран його найдовше слово.
36	Ввести речення та вивести на екран всі його слова, довжина яких не перевищує 7 символів.
37	Ввести речення. Подвоїти входження кожного слова. Для відокремлення слів використовувати пробіли.
38	Видалити з рядка всі слова, що починаються з приголосної літери.
39	В заданому рядку підрахувати кількість букв С.
40	У заданому рядку видалити кожен п'ятий символ.

Контрольні запитання

1. Чим відрізняється поточна довжина рядка від його загальної довжини?
2. Як позначається кінець рядка?
3. Які бібліотечні функції визначені для змінних рядкового типу?
4. Як порівняти рядки?
5. Як здійснити доступ до елемента рядка?
6. У чому особливість застосування функції розкладання рядка на лексеми?

Корисні посилання

1. Рядки. URL: <https://acode.com.ua/urok-82-ryadky-c-style/>
2. Основні функції для роботи з рядками. URL: <https://cplusplus.com/reference/cstring/>
3. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>

Лабораторне заняття № 7. Процедурне програмування

Мета: ознайомитись з видами функцій в С та набути практичних навичок з розроблення функцій.

План

1. Поняття про функцію та її складові.
2. Функції типу void.
3. Передавання параметрів у функцію.
4. Рекурсивні функції.
5. Завдання до лабораторної роботи.
6. Контрольні запитання.
7. Корисні посилання.

7.1 Поняття про функцію та її складові

Функція – поіменована послідовність описів та операторів, яка виконує деяку скінченні послідовність дій. Будь-яка функція складається із заголовка (оголошення функції) і тіла (визначення функції).

Виклик функції – зазначення ідентифікатора функції (її імені), за яким в круглих дужках наведено список *аргументів*, розділених комами.

Параметри, що вказуються в заголовку функції при її описі – **формальні параметри**

Параметри, що вказуються при виклику функції, називаються **фактичними параметрами (аргументами)**. На рисунку 7.1 фактичними параметрами, наприклад, є числа 52,5, 123,5, до яких застосовується функція **max**.

На рисунку 7.1 наведено приклад послідовного виклику декількох функцій.

```
double maxValue = max(52.5, 123.5);  
int j = swap(10, 2);  
func();
```

Рисунок 7.1 – Виклик функції

7.2 Функції типу void

Функції типу **void** – ті, що не повертають значення – це певний різновид команд, реалізований особливими програмними операторами. Оператор **func()** (див. рис. 7.1) виконує функцію **void func()**, тобто передасть керування функції, поки не виконаються всі її оператори. Коли функція поверне керування в основну програму, тобто завершить свою роботу, основна програма продовжить своє виконання з тієї позиції, де розташовується наступний оператор за оператором **func()**.

На рисунку 7.2 наведено приклад організації та виклику функції для обчислення факторіалу числа, яка не повертатиме в головну програму жодного значення.

```
#include <stdio.h>
#include <math.h>

void faktorial(int); //прототип функції

int main()
{
    int number; //змінна для збереження введеного значення
    printf(" What factorial should be calculated? ");
    scanf_s("%d", &number);
    faktorial(number); //виклик функції обчислення факторіала
    return 0;
}
//визначення функції
void faktorial(int value) //заголовок функції
{
    int result = 1;
    for (int i = 1; i <= value; i++)
    {
        result *= i;
    }
    printf("%d! = %d", value, result);
}
```

Рисунок 7.2 – Приклад функції типу **void**

Якщо функція повертає значення типу , то її виклик слід організовувати так, щоб значення, яке повертається, не використовувалося. Тобто, таку функцію не використовують у правій частині виразу.

7.3 Передавання параметрів у функцію

Існує два способи передавання параметрів у функцію: за значенням і за адресою.

```
#include <stdio.h>

void f(int i, int* j, int &k);

int main() {

    int i = 1, j = 2, k = 3;
    printf("i = %d, j = %d, k = %d\n", i, j, k); // Результат 1 2 3
    f(i, &j, k);
    printf("i = %d, j = %d, k = %d", i, j, k); //Результат 1 3 4
    return 0;
}
```

```
void f(int i, int* j, int& k) {
    i++; (*j)++; k++;
}
```

Рисунок 7.3 – Передавання параметрів у функцію

Перший параметр (**i**) передається за значенням. Його зміна в функції не впливає на вихідне значення. Другий параметр (**j**) передається за адресою за допомогою покажчика, при цьому для передавання у функцію адреси фактичного параметра використовується операція взяття адреси, а для отримання його значення в функції потрібна операція розіменування. Третій параметр (**k**) передається адресою за допомогою посилання.

Примітка. Для мови програмування C передавання параметрів у функцію має здійснюватись або за значенням або за покажчиком.

Розглянемо, яким чином можна передавати масив у функцію.

Приклад 7.1 Написати програму, в якій генерується одновимірний масив випадкових чисел та обчислюється сума кубів елементів масиву.

Дана програмна реалізація побудована таким чином: створюється динамічний масив **massive**, який ініціалізується випадковими елементами в діапазоні від 0 до 19. Потім здійснюється виклик функції **Sum(massive, N)** для обчислення суми кубів елементів масиву. Масив передається в функцію за адресою. Також, слід звернути увагу, на те, що в програмі реалізовано дві користувацькі функції. Перша **Cube(int x)** – обчислює та повертає куб числа. Друга функція **Sum(int* arr, int lenght)** – по-перше, викликає функцію для обчислення куба елемента масива **Cube(arr[i])**, куди передає по черзі всі елементи масиву, а потім обчислює суму кубів цих елементів (див. рис. 7.4).

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
//Прототипи функцій
long int Cube(int); // оголошення функції, яка повертатиме куб
елементу
long int Sum(int* arr, int lenght); // передаємо до аргументів
функції покажчик на масив та кількість його елементів

int main()
{
    int N;
    printf(«Enter number of elements in massive \n»);
    scanf_s(«%d», &N);
    int* massive = new int[N];
    printf(«Random generated massive is ...\n»);
    srand(time(NULL));
    //рандомна генерація елементів масиву
    for (int i = 0; i < N; i++) {
```

```

        massive[i] = rand() % 20; //кожен елемент масиву -
        випадкове число від 0 до 19
        printf(«%3d», massive[i]);
    }
    long int Suma = Sum(massive, N); //виклик функції
    printf(«\n Suma = %3d», Suma);
    return 0;
}

long int Cube(int x) //визначення функції, яка повертатиме куб
числа
{
    long int y = pow(x, 3);
    return y;
}

long int Sum(int* arr, int lenght) //визначення функції, яка
обчислюватиме суму кубів
{
    long int S = 0;
    for (int i = 0; i < lenght; i++) {
        S += Cube(arr[i]); //виклик функції Cube для піднесення до
        третього степеня кожного елемента масиву
    }
    return S;
}

```

Рисунок 7.4 – Передавання у функцію одновимірного масиву та обробка його елементів

Приклад 7.2 Розробити програму для пошуку максимального елемента двовимірного масива.

В даній програмній реалізації слід звернути увагу на декілька моментів. По-перше створення динамічного двовимірного масиву **mmm**: спочатку виділяється пам'ять для стовпця вказівників на рядки матриці (допоміжний масив), далі – в циклі виділяється пам'ять для вказівників на рядки матриці (основний масив). Для обчислення максимуму матриці реалізовано функцію **int maxArrayOfArray(int** a, int size1, int size2)**, яка приймає такі параметри: покажчик на масив, кількість рядків, кількість стовпців матриці. Через покажчик другого порядку можна передати масив у функцію тільки, якщо він динамічний. Виклик функції відбувається в основній програмі так: **maxArrayOfArray(mmm, H, W)**, а оскільки ім'я масиву – це покажчик на його початок, то «&» не потрібен. Вивільнення пам'яті відбувається в оберненому порядку відповідно до виділення пам'яті під масив.

```

#include <stdlib.h>
#include <stdio.h>
int maxArrayOfArray(int** a, int size1, int size2); //прототип
функції для пошуку максимального значення

```

```

int main()
{
    int H, W; //кількість рядків та стовпців відповідно
    scanf_s("%d %d", &H, &W);
    int** mmm = (int**)malloc(sizeof(int*) * H); //виділяємо
    пам'ять для стовпця вказівників на рядки матриці
    for (int i = 0; i < H; i++) {
        mmm[i] = (int*)malloc(sizeof(int) * W); //виділяємо
    пам'ять для вказівників на кожний рядок
    }
    //заповнення масиву
    for (int i = 0; i < H; i++) {
        for (int j = 0; j < W; j++) {
            mmm[i][j] = j + i;
            printf("%4d", mmm[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    //виклик функції та передавання параметрів
    int MaximumOfArray = maxValueOfArray(mmm, H, W);
    printf("\nMaximal element of array is %d", MaximumOfArray);
    //вивільнення пам'яті
    for (unsigned int i = 0; i < H; i++) {
        free(mmm[i]);
    }
    free(mmm);
    return 0;
}

//Оголошення функції
int maxValueOfArray(int** a, int size1, int size2) {
    int max = a[0][0];
    for (int i = 0; i < size1; i++) {
        for (int j = 0; j < size2; j++) {
            if (a[i][j] > max)
                max = a[i][j];
        }
    }
    return max;
}

```

Рисунок 7.5 – Обчислення максимуму двовимірного масиву

7.4 Рекурсивні функції

Рекурсія – метод визначення функції через її попередні та раніше визначені значення, а також спосіб організації обчислень, при якому функція викликає сама себе з іншим аргументом

Глибина рекурсії – максимальна кількість копій рекурсивної функції, що одночасно може знаходитись в пам'яті комп'ютера

Рекурсивна функція обов'язково повинна містити *умову завершення*, інакше вона буде нескінченною.

Приклад 7.3 Розробити рекурсивну функцію для обчислення факторіала цілого числа.

Повне рекурсивне визначення факторіала має такий вигляд:

$$n! = \begin{cases} 1, & \text{якщо } n = 0 \\ (n-1)! * n & \text{якщо } n > 0 \end{cases}$$

Рекурентне співвідношення для обчислення факторіала позначається так:
 $n! = n * (n-1)!$

Умова зупинки рекурсії: якщо $n=0$ чи $n=1$, то факторіал дорівнює 1.

Тоді рекурсивна функція обчислення факторіала цілого числа має такий вид:

```
int RecFact(int n)
{
    if (n == 0 || n == 1) return 1;
    else return (n * RecFact(n - 1));
}
```

Рисунок 7.6 – Рекурсивна функція обчислення факторіала

Наведемо декілька коментарів до цієї реалізації. Нехай потрібно обчислити факторіал числа 3. При виклику функції **RecFact(3)** буде викликана функція **RecFact(2)**, яка у свою чергу викличе **RecFact(1)**. Для останньої функції виконається умова зупинки рекурсії ($n==1$), у функцію **RecFact(2)** повернеться значення 1 і обчислиться значення 2 ($2!=2$), яке у свою чергу повернеться до функції **RecFact(3)** і буде використане для обчислення значення $3!$. Схематично цей процес можна позначити так:

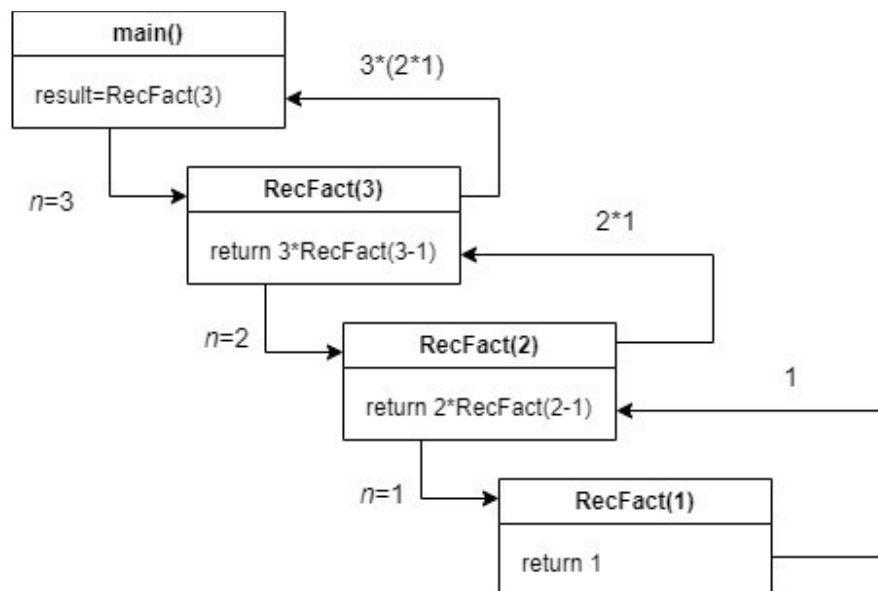


Рисунок 7.7 – Схема обчислення $3!$

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №7

Відповідно до варіанта (визначається за списком академгрупи) виконати 3 завдання. В кожній програмі реалізувати нескінченний цикл. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконані завдання 1, 2, 3;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта.
3. Умови завдань 1-3, тексти та результати роботи програм.
4. Відповіді на контрольні запитання.

Зауваження. Створені функції перенести в окремий .с-файл, а прототипи – зазначити в окремому .h-файлі.

Варіанти завдань до лабораторної роботи №7

Завдання 1. В лабораторній роботі №4 (завдання 2) визначити відповідну функцію (або функції) для роботи з масивами. Розроблені функції (функція) мають виконувати одну певну дію.

Завдання 2. Написати програму відповідно до варіанта завдання

№ вар.	Завдання
1	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{5} + 5}{2} + \frac{\sqrt{12} + 12}{2} + \frac{\sqrt{19} + 19}{2}$
2	Обчислити значення y , визначивши та використавши необхідну функцію: $x = \frac{1 + \sin 1}{3} + \frac{3 + \sin 3}{3} + \frac{5 + \sin 5}{3}$
3	Розрахувати значення $z = \max(a, 2b) + \max(2a - b, b)$ визначивши та використавши функцію $\max(x, y)$ – максимальне з двох чисел.
4	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{5} + 5}{\sqrt{7} + 7} + \frac{\sqrt{12} + 12}{\sqrt{8} + 8} + \frac{\sqrt{19} + 19}{\sqrt{2} + 2}$
5	Обчислити значення y , визначивши та використавши необхідну функцію: $y = \frac{2 + \sin 2}{\sin 5 + 5} + \frac{6 + \sin 6}{\sin 3 + 3} + \frac{1 + \sin 1}{\sin 4 + 4}$
6	Розрахувати значення $z = \min(2a, b + a) + \min(2a - b, b)$ визначивши та використавши функцію $\min(x, y)$ – мінімальне з двох чисел

№ вар.	Завдання
7	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{8} + 15}{\sqrt{15} + 8} + \frac{\sqrt{6} + 12}{\sqrt{12} + 6} + \frac{\sqrt{7} + 21}{\sqrt{21} + 7}$
8	Дано трикутник ABC довжинами своїх сторін a, b, c . Знайти його медіани, створивши функцію обчислення медіани $0,5\sqrt{2b^2 + 2c^2 - a^2}$ (для сторони a). Сторони a, b, c ввести з клавіатури.
9	Розрахувати значення $z = \text{sign}(x) + \text{sign}(y)$, визначивши та використавши функцію $\text{sign}(a)$, яка приймає такі значення: $\text{sign}(a) = -1$ при $a < 0$, $\text{sign}(a) = 0$ при $a = 0$, $\text{sign}(a) = 1$ при $a > 0$.
10	Дано основу та висоту двох рівнобічних трапецій. Знайти суму їхніх периметрів. Визначити функцію для розрахунку периметра рівнобічної трапеції за її основою та висотою.
11	Написати програму визначення периметра трикутника, який задано координатами його вершин. Довжину сторони визначати за допомогою функції.
12	Написати програму визначення площі п'ятикутника із заданими координатами його вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)$ як суму площ трьох трикутників. Площу трикутника, визначену за координатами вершин, розрахувати за допомогою функції.
13	Задано 8 різних чисел. Визначити максимальне з них, використовуючи функцію визначення максимального з двох чисел.
14	Знайти всі двозначні прості числа, визначивши функцію для обчислення простих чисел
15	Знайти значення виразу $\frac{2 \cdot 5! + 3 \cdot 8!}{6! + 4!}$, де $n!$ – факторіал. Визначити ітераційну функцію для розрахунку факторіала.
16	Знайти площу прямокутного трикутника ABC, який задано координатами вершин. Довжину сторони обчислювати за допомогою функції.
17	Написати програму визначення площі шестикутника із заданими координатами його вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6)$ як суму площ чотирьох трикутників. Площу трикутника, сторони якого визначено за координатами вершин, розрахувати за допомогою функції.
18	Знайти суму цифр двох цілих двозначних чисел a і b , створивши функцію визначення цифр числа.
19	Знайти значення виразу $\text{sh}(x) \tan(x + 1) - \text{ctg}^2(2 + \text{sh}(x - 1))$, визначивши функцію $\text{sh}(x) = \frac{e^x - e^{-x}}{2}$. Значення x ввести в клавіатури.
20	Дано масив m цілих чисел. Написати програму визначення середнього арифметичного цих чисел, використовуючи функцію.
21	Дано масив цілих чисел A . Знайти суми додатніх і від'ємних елементів масива, використовуючи функцію визначення суми.
22	Розрахувати значення $z = \text{modd}(x) - \text{modd}(y)$, визначивши та використавши функцію $\text{modd}(a)$, яка приймає такі значення: $\text{modd}(a) = a$ при $a \geq 0$, $\text{modd}(a) = -a$ при $a < 0$.
23	Трикутник задано координатами його вершин. Визначити вид трикутника (рівнобедрений, рівносторонній, прямокутний чи звичайний). Довжину сторони визначати за допомогою функції.
24	Обчислити значення y визначивши та використавши необхідну функцію: $y = \frac{1 + \sin 4}{\sin 1 + 4} + \frac{7 + \sin 5}{\sin 7 + 5} + \frac{3 + \sin 2}{\sin 3 + 2}$

№ вар.	Завдання
25	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{7} + 13}{\sqrt{13} + 7} + \frac{\sqrt{12} + 15}{\sqrt{15} + 12} + \frac{\sqrt{21} + 32}{\sqrt{32} + 21}$
26	Обчислити значення y , визначивши та використавши необхідну функцію: $y = \frac{2 + \sin 3}{\sin 2 + 3} + \frac{1 + \sin 5}{\sin 1 + 5} + \frac{4 + \sin 7}{\sin 4 + 7}$
27	Дано три квадратних рівняння: $ax^2 + bx + c$, $bx^2 + ax + c$, $cx^2 + ax + b$. Скільки з них мають дійсні корені. Визначити функцію, яка розпізнає наявність дійсних коренів у квадратному рівнянні.
28	Дано натуральне число n та цілі числа $a_1, a_2, \dots, a_n$. Знайти кількість a_i ($i=1, 2, \dots, n$), які є повними квадратами. Визначити функцію, яка розпізнає повні квадрати.
29	Знайти всі тризначні прості числа. Визначити функцію, яка розпізнає прості числа.
30	Дано два натуральні числа. З'ясувати, в якому з них сума цифр більше. Визначити функцію для розрахунку суми цифр числа.
31	Дано шість різних чисел. Визначити максимальне з них. Визначити функцію, яка знаходить максимум з двох чисел
32	Дано два натуральних числа. Знайти їхнє найменше спільне кратне, використовуючи функцію, яка реалізує алгоритм Евкліда.
33	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{512} + 512}{5} + \frac{\sqrt{2} + 2}{5} + \frac{\sqrt{19} + 19}{5}$
34	Обчислити значення y , визначивши та використавши необхідну функцію: $x = \frac{5 + \sin 5}{3} + \frac{3 + \sin 3}{12} + \frac{10 + \sin 10}{7}$
35	Розрахувати значення $z = \min(2a, b) + \min(a - 2b, b)$ визначивши та використавши функцію $\min(x, y)$ – мінімальне з двох чисел.
36	Обчисліть значення x , визначивши та використавши необхідну функцію: $x = \frac{\sqrt{10} - 10}{\sqrt{7} + 7} + \frac{\sqrt{7} - 7}{\sqrt{8} + 8} + \frac{\sqrt{21} - 21}{\sqrt{2} + 2}$
37	Обчислити значення y , визначивши та використавши необхідну функцію: $y = \frac{2 + \cos 2}{\sin 5 + 5} + \frac{6 + \cos 6}{\sin 3 + 3} + \frac{1 + \cos 1}{\sin 4 + 4}$
38	Розрахувати значення $z = \max(2a, b - a) + \max(2a + b, b)$ визначивши та використавши функцію $\text{Max}(x, y)$ – максимальне з двох чисел
39	Знайти значення виразу $\text{ch}(x) \tan(x - 1) + \text{ctg}^2(2 - \text{ch}(x + 1))$, визначивши функцію $\text{ch}(x) = \frac{e^x + e^{-x}}{2}$. Значення x ввести в клавіатури.
40	Задано 6 різних чисел. Визначити мінімальне з них, використовуючи функцію визначення мінімального з двох чисел.

Завдання 3. Написати програму відповідно до свого варіанта завдання.

- В одній програмі створити функції для обчислення арифметичного виразу ітераційним та рекурсивним способами. Результати рекурсивного та ітераційного способу повинні співпасти. Всю необхідну функціям інформацію слід передавати лише за допомогою параметрів.

- Використання глобальних змінних у підпрограмах не допускається. Інформація з підпрограми повинна передаватися лише за допомогою параметра. Виведення результатів роботи підпрограм повинне виконуватися в головній програмі. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями.

№	Завдання	№	Завдання	№	Завдання
1	$\sum_{i=k}^N \frac{i^2}{k^2 + N^2}$	11	$\sum_{i=1}^N \frac{\cos(\sin i)}{1 + \sin^2 i}$	21	$\sum_{i=N}^{22} \frac{\sqrt{i^2 + N^2}}{i}$
2	$\sum_{k=N}^{19} \sqrt{\sin^2(k) + \cos^2 \frac{N}{k}}$	12	$\sum_{i=3}^N \frac{\cos\left(\frac{i}{2}\right) + \sin 2i}{1 + \sin(\cos i)}$	22	$\prod_{i=N}^{10} \left(\frac{i + \frac{1}{i^2}}{\sqrt{1 + e^{\frac{1}{i}}}} \right)$
3	$\prod_{i=1}^{15} \frac{\sin^2(i) + \cos^2\left(\frac{1}{i}\right)}{i^2}$	13	$\prod_{k=N}^{19} \left(\frac{k - N}{k + N} + 1 \right)$	23	$\prod_{k=1}^N \left(\frac{k^2}{N^2} \cos \frac{N^2}{k^2} \right)$
4	$\sum_{k=1}^N \frac{1}{(2k + 1)^2}$	14	$\prod_{k=N}^{15} \sqrt{\sin^3 k + \operatorname{ctg}^2 \frac{N}{k}}$	24	$\sum_{i=1}^N \frac{\sin i}{1 + \cos i}$
5	$\sum_{i=1}^{15} \frac{\sin(10i) + \cos\left(\frac{10}{i}\right)}{\sqrt{i}}$	15	$\sum_{i=1}^N \frac{1 + \cos^2(\sin i)}{\sqrt{\sin^2 \cos i + 1}}$	25	$\sum_{i=k}^N \frac{\sin(\cos i)}{1 + \cos^2 i}$
6	$\sum_{k=1}^N \frac{1 + \sqrt{1 + k^2}}{k^2}$	16	$\sum_{i=1}^N \left(\frac{1}{i} + \sqrt{1 + \sin^2 i} \right)$	26	$\sum_{i=N}^{20} \frac{\sin i + \cos i}{1 + \cos i \sin i}$
7	$\prod_{k=1}^N \sqrt{1 + \frac{N}{k}}$	17	$\sum_{k=1}^N \frac{1 + \frac{i}{N}}{i^2}$	27	$\sum_{i=1}^N \frac{\sqrt{1 + \cos^2 i} \sin i}{1 + \cos i \sin^2 i}$
8	$\sum_{j=2}^N \frac{j(N - j)}{j^2 + (N - j)^2}$	18	$\prod_{k=1}^N \left(\frac{k - N}{k + N} + 1 \right)$	28	$\prod_{j=1}^{15} \left(\frac{iN}{i^2 + N^2} \right)$
9	$\sum_{i=k}^{15} \frac{\cos(i)}{1 + \sin^2(i)}$	19	$\prod_{i=K}^N \left(\frac{K}{i} + \frac{i}{N} \right)$	29	$\prod_{j=2}^N \left(\frac{j^3(N - j)}{j^2 + \sqrt{N - j}} \right)$
10	$\sum_{i=k}^N \frac{\sin(i) \cdot \cos(i)}{1 + \sin^2(i)}$	20	$\prod_{i=N}^{16} \left(\frac{iN}{i^2 + N^2} \right)$	30	$\sum_{k=1}^N \sin \frac{k^2}{N} \cos \frac{N^2}{k^{N-k}}$
31	$\sum_{k=1}^{\infty} \frac{k + 1}{k^3 - k^2 + 1}$	32	$\sum_{k=1}^{\infty} \frac{k + 1}{k^3 + 1}$	33	$\sum_{k=1}^{\infty} \frac{x^{2k+1}}{(2k + 1)(2k - 1)}$
34	$\sum_{k=1}^{\infty} \frac{x^k \cdot \sin\left(\frac{2\pi k}{3}\right)}{k}$	35	$\sum_{k=1}^{\infty} \frac{\cos(2k - 1)}{(2k - 1)^2}$	36	$\sum_{k=1}^{\infty} \frac{\sin(k - 1)}{(2k - 1)}$
37	$\sum_{k=1}^{\infty} \frac{1}{4^k}$	38	$\sum_{k=1}^{\infty} \frac{2^k}{k(2^k + 1)}$	39	$\sum_{k=1}^{\infty} (-1)^k \frac{3k - 1}{k^3 + 1}$
40	$k + \sum_{k=1}^{\infty} \frac{1}{k}$	41	$\sum_{i=k}^N \frac{\sin(i)}{1 + \sin^2(i)}$	42	$\sum_{i=0}^N \frac{1}{2^i}$

Контрольні запитання

1. Які переваги дає використання підпрограм (користувачьких функцій)?
2. Які різновиди функцій є у мовах C та C++?
3. У чому полягає відмінність формальних параметрів від аргументів?
4. Як повертаються значення з функції? Скільки значень можна повернути?
5. Чи може програміст назвати свою функцію **sin()** ?
6. Що таке рекурсивна функція?
7. Як визначається глибина рекурсії?

Корисні посилання

1. Рекурсія і числа Фібоначчі. URL: <https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/#toc-5>
2. Стек і купа. URL : <https://acode.com.ua/urok-111-stek-i-kupa/>
3. Створення власних заголовних файлів URL: <https://www.youtube.com/watch?v=7XKwhT9Mf1Y>
4. Підключення та використання бібліотек в Visual Studio. URL: <https://acode.com.ua/pidklyuchennya-ta-vykorystannya-bibliotek-v-visual-studio/#toc-2>
5. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
6. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html>

Лабораторне заняття № 8. Робота з файлами

Мета: ознайомитись з типами файлів та набути практичних навичок опрацювання файлів у програмах.

План

1. Основні функції для роботи з файлами в С.
3. Запис та читання інформації з файлу.
4. Завдання до лабораторної роботи.
5. Контрольні запитання.
6. Корисні посилання.

8.1 Основні функції для роботи з файлами в С.

Основні функції файлового введення/виведення оголошено у заголовному файлі **stdio.h**. Серед іншого, в цьому ж файлі описано значення NULL, EOF тощо.

Таблиця 8.1 – Основні функції файлової системи С

Функція	Призначення
fopen()	Відкрити файл
fclose()	Закрити файл
putc()	Записати символ у файл
fputc()	Записати символ у файл
getc()	Зчитати символ із файла
fgetc()	Зчитати символ із файла
fputs()	Записати рядок у файл
fseek()	Встановити вказівник поточної позиції на певний байт файла
ftell()	Повернути значення вказівника поточної позиції у файлі
fprintf()	Друкувати у файл
fscanf()	Зчитувати з файлу
feof()	Повернути значення true , якщо досягнуто кінець файла
ferror()	Повернути значення true , якщо трапилася помилка
rename()	Перейменувати фізичний файл
remove()	Знищити фізичний файл

8.2 Запис та читання інформації з файлу

Наведемо декілька прикладів читання та запису інформації у файл.

Приклад 8.1 Створити текстовий файл та записати в нього ціле число. Програма реалізація матиме такий вид:

```
#include <stdio.h>
#include <stdlib.h>
```

```

int main() {
    int num;
    FILE* fptr;
    fopen_s(&fptr, "C:\\temp\\program.txt", "a");
    if (fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    printf("Enter num: ");
    scanf_s("%d", &num);

    fprintf(fptr, "%d", num);
    fclose(fptr);

    return 0;
}

```

Рисунок 8.1 – Запис цілого числа у файл

Наведемо декілька коментарів до цієї програми. По-перше, дану програмну реалізацію виконано в середовищі MS VisualStudio, а тому для відкриття файлу використано функцію **fopen_s** з додатковим аргументом **&fptr**. Також, оскільки файл розташовано не в робочому каталозі, то для роботи з ним вказано повний шлях. Далі відбувається перевірка, чи знайдено файл з таким ім'ям за вказаною адресою. Якщо файл не знайдено, то відбувається завершення програми. Якщо файл знайдено, то відбувається запит на введення цілого числа з клавіатури, а потім запис цього числа у файл. Після цього файл закривається.

Приклад 8.2 Прочитати інформацію, яка міститься в текстовому файлі. Програмна реалізація матиме такий вигляд:

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num;
    FILE* fptr;
    if ((fptr = fopen("C:\\temp\\program.txt", "r")) == NULL) {
        printf("Error! opening file");
        exit(1);
    }
    fscanf (fptr, "%d", &num);
    printf("Value of n=%d", num);
    fclose(fptr);

    return 0;
}

```

Рисунок 8.2 – Зчитування інформації з файлу

Наведемо декілька коментарів до цієї програми. Для відкриття файлу використано функцію **fopen** з параметром **r**. Другий параметр – повний шлях до файлу. Після перевірки існування та можливості відкриття файлу відбувається зчитування інформації, далі – виведення на екран. Закриття файлу реалізовано через функцію **fclose**.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №8

Відповідно до варіанта (визначається за списком академгрупи) виконати 1 завдання. Умови та варіанти завдань наведено нижче.

До захисту лабораторної роботи представити:

- виконане завдання 1;
- звіт з виконання лабораторної роботи (шаблон на сторінці курсу в moodle);
- відповіді на запитання викладача по завданням.

Структура звіту з лабораторної роботи:

1. Титульна сторінка.
2. Номер варіанта.
3. Умова завдання 1, текст та результати роботи програми.
4. Відповіді на контрольні запитання.

Варіанти завдань до лабораторної роботи №8

Завдання 1. Розробити програму для свого варіанта. У програмі має бути реалізовано:

- меню користувача, яке надає можливість вносити записи у файл, виводити на екран вміст файлу, завершити сеанс. Реалізувати через нескінченний цикл та оператор **switch**;

- програма має записувати або зчитувати та виводити на екран таблицю з назвами відповідних стовпців. Кожен запис записати в окремий рядок, а числові дані зчитувати у відповідному форматі.

№ вар.	Завдання
1	Інформація про студентів, що включає поля з прізвищем та ім'ям, номером групи, балами за три навчальні дисципліни. Знайти середній бал кожного студента. Виконувати пошук студентів певної групи, номер якої вводить користувач.
2	Інформація про розклад групи, включає такі поля: день тижня, назва навчальної дисципліни, номер аудиторії, прізвище викладача. Знайти всі предмети у вказаний користувачем день тижня, а також для вказаного предмету вивести дні тижня та аудиторії.
3	Інформація про вартість товарів у магазині (назва товару, ціна одиниці, кількість наявних одиниць, постачальник). Знайти сумарну вартість кожного товару та здійснити пошук товарів заданого постачальника.
4	Каталог автомобілів: модель, рік випуску, пробіг, об'єм двигуна. Обчислити середньорічний пробіг та знайти автомобілі із вказаним об'ємом двигуна.
5	Облік штрафів накладених на автомобілістів зберігає інформацію про прізвища інспектора, що вивписа штраф та оштрафованого громадянина, дату накладання штрафу, штрафну суму, текстове поле з описом порушення. Обчислити суму штрафів та виконувати пошук всіх штрафів після вказаної дати.
6	Туристична агенція зберігає інформацію про прізвище клієнта, дату та країну подорожі, а також вказано суму туристичної путівки та страховки. Обчислити середню вартість проданих туристичних путівок та пошук путівок, поданих після вказаної дати

№ вар.	Завдання
7	Пункт фотодруку зберігає ім'я замовника, дату замовлення, кількість замовлених фотографій та вартість замовлення. Обчислити сумарну кількість роздрукованих фотографій та виконати пошук замовлень після вказаної дати.
8	Страховий агент зберігає прізвище клієнта, код угоди, дату укладання угоди, суму страховки та поле, яке вказує, чи відбувся страховий випадок за період договору чи ні. Знайти всі договори, за якими відбувалися страхові випадки. Обчислити середню суму страховки за договором.
9	Ломбард зберігає такі дані: назву предмету, прізвище власника, виплачену суму, дату укладання договору, ставку відсотку та відмітку про повернення речі. Знайти всі предмети, які ще не повернено власнику. Обчислити сумарну виплачену суму після вказаної дати.
10	В аптечному файлі є такі дані: назва лікарського засобу, ціна закупівлі цього засобу, торговельну надбавку та дату поставки. Вивести на екран всі лікарські засоби, які поставлено після вказаної дати. Обчислити та надрукувати для кожного засобу роздрібну ціну.
11	Диспетчер таксі зберігає інформацію про дату виклику, водія, що обслуговував виклик, кілометраж та ціну за 1 км за цим маршрутом, а також поле, що показує успішність виклику (відбувся чи ні). Обчислити сумарну кількість оплачених клієнтами коштів. Знайти всі виклики, що обслуговував заданий водій.
12	«Зоопарк»: загальна кількість тварин; кількість тварин кожного виду, місто розташування зоопарку, назва зоопарку, кількість робітників. Вивести відомості про зоопарки, в яких є тигри.
13	«Програміст»: прізвище, ім'я, по-батькові, стать, дата народження (рік, місяць, число), освіта, номер телефону. Вивести відомості про програмістів, яким менше 25 років.
14	«Автомобіль»: марка; колір; серійний номер; рік випуску; рік техогляду; ціна. Вивести дані про автомобілі, яким більше двох років.
15	«Студент»: прізвище; ім'я; по-батькові; стать; зріст; вага; дата народження (рік, місяць, число); номер телефону; ВНЗ; курс; група; середній бал; спеціальність. Вивести відомості про всіх студентів, у яких середній бал нижче 70.
16	«Людина»: прізвище; ім'я; по-батькові; стать; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашня адреса (поштовий індекс, місто, вулиця, будинок, квартира). Вивести відомості про наймолодшу людину.
17	«Школяр»: прізвище; ім'я; по-батькові; стать; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашня адреса (поштовий індекс, місто, вулиця, будинок, квартира); школа; клас. Вивести відомості про всіх учнів п'ятих класів.
18	«Покупець»: прізвище; ім'я; по-батькові; стать; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашня адреса (поштовий індекс, місто, вулиця, будинок, квартира); номер кредитної картки. Вивести дані про покупців з міста Запоріжжя.
19	«Робітник»: прізвище; ім'я; по-батькові; стать; дата народження (рік, місяць, число); номер телефону; домашня адреса (поштовий індекс, місто, вулиця, будинок, квартира); № цеху; табельний номер; освіта; рік вступу на роботу. Вивести дані про робітників, які вступили на роботу у 2020 році.
20	«Програміст»: прізвище; ім'я; по-батькові; стать; дата народження (рік, місяць, число); номер телефону; освіта. Вивести дані про програмістів, яким менше 25 років.
21	«Науковець»: прізвище; ім'я; по-батькові; стать; дата народження (рік, місяць, число); посада; науковий ступінь; номер телефону; домашня адреса (поштовий індекс, місто, вулиця, будинок, квартира). Вивести на екран відомості про науковців кандидатів технічних наук.

№ вар.	Завдання
22	«Футболіст»: прізвище; ім'я; по-батькові; дата народження (рік, місяць, число); номер телефону; назва команди; результативність (кількість голів); кількість ігор. Вивести на екран відомості про футболістів, які провели за свою команду більше 50 матчів.
23	«Міжнародна компанія»: назва; інтернет-сайт; адреса головного офісу (поштовий індекс, країна, місто, вулиця, будинок); тривалість перебування на світовому ринку; кількість співробітників; кількість філіалів в Європі. Вивести на екран дані про міжнародні компанії, кількість співробітників яких перевищує 10000.
24	«Програмне забезпечення»: назва; назва компанії виробника; рік виходу; ціна. Вивести дані про програмне забезпечення, яке дорожче за 2000 грн.
25	«Мультфільм»: назва; режисер (прізвище; ім'я); рік виходу; країна; вартість; прибуток. Вивести дані про мультфільми компанії «Walt Disney».
26	«Баскетболіст»: прізвище; ім'я; по-батькові; зріст; вага; дата народження (рік, місяць, число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість очок); кількість ігор. Вивести на екран відомості про баскетболістів, які закинули за свою команду більше 100 очок.
27	«Область»: назва; обласний центр; населення; площа території; губернатор. Вивести дані про області, населення яких менше 2 млн. жителів.
28	«Мотоцикл»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про мотоцикли марки «Harley-Davidson».
29	«Країна»: назва; столиця; державна мова; населення; площа території; грошова одиниця; голова держави. Вивести дані про держави, населення яких більше 20 млн. жителів.
30	«Рейс» марка автомобіля; номер автомобіля; пункт призначення; вантажопідйомність (в тонах); вартість одиниці вантажу; загальна вартість вантажу. Вивести дані про автомобілі, вантажопідйомність яких більше 2 тон.
31	Дані про книги домашньої бібліотеки записано в N рядках символів, кожен з яких містить такі дані: автор, назва книги, видавництво, рік видання. Підрахувати кількість книг заданого автора.
32	Дані про книги домашньої бібліотеки записано в N рядках символів, кожен з яких містить такі дані: автор, назва книги, видавництво, рік видання. Підрахувати кількість книг виданих після 2000 року.
33	Дані про клієнтів банку записано в N рядках символів, кожен з яких містить такі дані: прізвище, номер рахунку, сума вкладу, дата. Обчислити загальну суму вкладів, що є на рахунках всіх клієнтів.
34	Дані про книги домашньої бібліотеки записано в N рядках символів, кожен з яких містить такі дані: автор, назва книги, видавництво, рік видання. Підрахувати кількість книг заданого видавництва.
35	Дані про успішність студентів записано в N рядках символів, кожен з яких містить такі дані: прізвище, ім'я, оцінки з п'яти предметів (у форматі відмінно, добре, задовільно або в транслітерації). Підрахувати кількість студентів, які вчаться на «добре» та «відмінно».
36	Дані про автомобілі записано в N рядках символів, кожен з яких містить такі дані: марка авто, номер, колір, прізвище власника, відмітка про техогляд. Підрахувати кількість автомобілів заданої марки, у яких протермінований техогляд.
37	Дані про продукцію на складі записано в N рядках символів, кожен з яких містить такі дані: назва виробу, код виробу, кількість виробів, ціна одного виробу, дата. Знайти вартість продукції виготовленої після заданої дати.
38	Дані про програмне забезпечення записано в N рядках символів, кожен з яких містить такі дані: назва, назва компанії виробника, рік виходу, ціна. Вивести дані про

№ вар.	Завдання
	програмне забезпечення, ціна якого більше 2000 грн.
39	Дані про мультиплікаційні фільми записано в N рядках символів, кожен з яких містить такі дані: назва, режисер, рік виходу, країна, прибуток, компанія. Вивести дані про мультфільми компанії «Walt Disney».
40	Ломбард зберігає такі дані: назву предмету, прізвище власника, виплачену суму, дату укладання договору, ставку відсотку та відмітку про повернення речі. Знайти всі предмети, які ще не повернено власнику. Обчислити сумарну виплачену суму після вказаної дати.

Контрольні запитання

1. Які переваги дає використання підпрограм (користувачьких функцій)?
2. Які різновиди функцій є у мовах C та C++?
3. У чому полягає відмінність формальних параметрів від аргументів?
4. Як повертаються значення з функції? Скільки значень можна повернути?
5. Чи може програміст назвати свою функцію **sin()** ?
6. Що таке рекурсивна функція?
7. Як визначається глибина рекурсії?

Корисні посилання

1. Файлове введення/виведення. URL: <https://en.cppreference.com/w/c/io>
2. Параметри доступу до файлу. URL: <https://en.cppreference.com/w/c/io/fopen>
3. C-файли. URL: https://w3schoolsua.github.io/c/c_files.html#gsc.tab=0
4. Відлагодження програм: степпінг і точки зупину. URL: <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
5. C++ Programming Style Guidelines. URL: <https://geosoft.no/development/cppstyle.html>

Рекомендована література

1. Ковалюк Т. В. Алгоритмізація та програмування : підручник. Львів : «Магнолія 2006», 2022. 400 с.
2. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1 : навчальний посібник. Львів : «Новий Світ – 2000», 2021. 337 с.
3. Козак Л. І., Костюк І. В., Стасевич С. П. Основи програмування : навчальний посібник. Львів : «Новий Світ – 2000», 2024. 328 с.
4. Кормен Т., Лейзерсон Ч., Рівест Р., Стайн К. Вступ до алгоритмів. Київ : К.І.С., 2023. 1288 с.
5. Мартін Роберт С. Чистий код : створення і рефакторинг за допомогою Agile. Харків : Ранок, 2020. 448 с.
6. Мартін Роберт С. Чистий кодер : створення і рефакторинг за допомогою Agile. Харків : Фабула, 2023. 256 с.
7. Томас Д., Хант Е. Програміст-прагматик: друге ювілейне видання. Київ : Науковий світ, 2024. 392 с.
8. Бхаргава А. Грокаємо алгоритми : ілюстрований посібник для програмістів і допитливих. Київ : ArtHuss, 2023. 256 с.
9. Richard M Reese. Understanding and Using C Pointers: Core Techniques for Memory Management. O'Reilly Media, 2020. 225 p.
10. Horton I., Gonzalez-Morris G. Beginning : From Beginner to Pro 7th ed. Edition. NY : Apress, 2024. 726
11. Відлагодження програм: степпінг і точки зупину. URL : <https://acode.com.ua/urok-29-vidlagodzhennya-program-stepping-i-tochky-zupynu/>
12. C++ Programming Style Guidelines. URL : <https://geosoft.no/development/cppstyle.html>
13. C-Підручник. URL : <https://w3schoolsua.github.io/c/index.html#gsc.tab=0>

Використана література

1. Ковалюк Т. В. Алгоритмізація та програмування : підручник. Львів : «Магнолія 2006», 2022. 400 с.
2. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1 : навчальний посібник. Львів : «Новий Світ – 2000», 2021. 337 с.
3. Мартін Роберт С. Чистий код : створення і рефакторинг за допомогою Agile. Харків : Ранок, 2020. 448 с.
4. Шпак З. Я. Програмування мовою С. Львів : Львівська політехніка, 2018. 436 с.
5. Кормен Т., Лейзерсон Ч., Рівест Р. Вступ до алгоритмів. Київ : К.І.С., 2019. 1296 с.
6. Бородкіна І., Бородкин Г. Інженерія програмного забезпечення. Київ : Центр навчальної літератури, 2018. 204 с.
7. Кнут Д. Мистецтво програмування. Київ : Кондор, 2018. 720 с.
8. Галісеєв Г. Системне програмування. Київ : Університет «Україна», 2019. 113 с.
9. Richard M Reese. Understanding and Using C Pointers: Core Techniques for Memory Management. O'Reilly Media, 2020. 225 p.

Навчально-методичне видання
(українською мовою)

Кривохата Анастасія Григорівна
Кудін Олексій Володимирович

ОСНОВИ ПРОГРАМУВАННЯ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Рецензент *О.С. Пиєнична*
Відповідальний за випуск *А.О. Лісняк*
Коректор *А.Г. Кривохата*