МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кафедра комп'ютерної інженерії

Дисципліна: Програмування
Спеціальність: 123 «Комп’ютерна інженерія»
Освітній рівень: Перший (бакалаврський)
Семестр: I

ЛАБОРАТОРНА РОБОТА № 2

Тема: Розгалужені та циклічні обчислювальні процеси у мові С, консольні меню та оформлення інженерної звітності

---

1. МЕТА РОБОТИ

- Набути практичних навичок проектування комбінованих алгоритмів із використанням операторів розгалуження (`if-else` , тернарного оператора) та оператора множинного вибору (`switch-case`).
- Дослідити алгоритмічні особливості та практичну реалізацію циклів із передумовою (`while`), післяумовою (`do-while`) та параметричних циклів (`for`).
- Опанувати чисельні методи наближеного обчислення функцій за допомогою степеневих рядів та рекурентних співвідношень.
- Закріпити навички системного контролю версій Git: робота з файлом `.gitignore` , семантичні коміти за стандартом Conventional Commits.
- Опанувати академічні стандарти підготовки технічної документації шляхом формування **обов'язкової пояснювальної записки у форматі PDF**.

---

2. НЕОБХІДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

1. Текстовий редактор / середовище розробки: **Visual Studio Code** (із розширенням *C/C++* від Microsoft) або **CLion**.
2. Компілятор: **GCC** (GNU Compiler Collection).

3. Система контролю версій: **Git**.
4. Програма для підготовки інженерної документації: **LaTeX**, **MS Word** або **LibreOffice Writer** із можливістю експорту в PDF.

---

3. ПОСЛІДОВНІСТЬ ВИКОНАННЯ РОБОТИ

Етап 0. Підготовка робочого середовища та Git

1. Відкрийте термінал і перейдіть до кореневого каталогу вашого репозиторію дисципліни:

   ```
   cd ~/projects/programming-sem1
   ```

2. Перевірте статус репозиторію за допомогою команди:

   ```
   git status
   ```

   Переконайтеся, що робоча директорія чиста, а файл `.gitignore` надійно ігнорує скомпільовані бінарні файли (`*.exe` , `*.out` , `*.o`).
3. Створіть новий підкаталог `lab02` для поточної роботи та перейдіть у нього:

   ```
   mkdir lab02
   cd lab02
   ```

Етап 1. Табулювання кусочно-неперервної функції (`task1.c`)

1. У каталозі `lab02` створіть файл `task1.c` .
2. Відповідно до свого номера варіанта з **Таблиці 1** реалізуйте програму, яка:
   - Запитує у користувача межі відрізка $[X_{start}, X_{end}]$ та крок табулювання dX (змінні типу `double`).
   - Здійснює обов'язкову валідацію вхідних параметрів: $X_{start} < X_{end}$ та $dX > 0$. Якщо параметри некоректні — програма виводить повідомлення про помилку та завершує роботу з ненульовим кодом повернення.
   - Використовує параметричний цикл (`for`) або цикл із передумовою (`while`) для проходження по всьому діапазону від X_{start} до X_{end} включно.
   - На кожній ітерації за допомогою конструкції `if-else` визначає, до якого інтервалу належить поточне значення X, і обчислює відповідне значення функції $Y(X)$.
   - Здійснює перевірку на допустимість математичних операцій: якщо в точці X відбувається ділення на нуль або під коренем виникає від'ємне число, програма не

повинна аварійно завершуватися — у таблиці замість значення має друкуватися повідомлення `[Не визначено]`.
- Форматує вивід результатів у вигляді чіткої текстової таблиці з фіксованою шириною колонок та вирівнюванням (використовуйте модифікатори формату, наприклад, `%10.2lf | %12.4lf`).

3. Скомпілюйте програму з контролем помилок:
```
gcc -Wall -Wextra task1.c -o task1 -lm
./task1
```
4. Зафіксуйте створений сирцевий файл у Git:
```
git add task1.c
git commit -m "feat(lab2): implement piecewise function tabulation"
```

Етап 2. Наближене обчислення функцій чисельним рядом (`task2.c`)

1. У каталозі `lab02` створіть файл `task2.c`.
2. Відповідно до свого номера варіанта з **Таблиці 2** реалізуйте програму наближеного обчислення нескінченного ряду, яка:
   - Зчитує значення аргументу X та бажану інженерну точність ε (наприклад, $\varepsilon = 10^{-5}$ або $\varepsilon = 10^{-7}$).
   - Перевіряє, чи потрапляє введений аргумент X в область збіжності вашого ряду. Якщо ні — повідомляє про неприпустиме значення.
   - **Математична оптимізація:** Обчислення кожного наступного доданка ряду a_n здійснює **виключно через рекурентне співвідношення** відносно попереднього члена a_{n-1}:
   $$a_n = a_{n-1} \cdot R(n, X)$$
   Категорично заборонено обчислювати факторіал або степінь "в лоб" через важкі функції бібліотеки `math.h` на кожній ітерації!
   - За допомогою циклу з передумовою `while` веде накопичення суми доти, доки поточний доданок за модулем не стане меншим за точність: $|a_n| < \varepsilon$. Для взяття модуля дійсного числа використовуйте функцію `fabs()` із `<math.h>`.
   - **Апаратний захист (Timeout Guard):** Реалізує захисний лічильник ітерацій (максимум $100\,000$). Якщо за цей ліміт точність не досягнута (ряд розбігається) — цикл примусово переривається командою `break` з виведенням попередження.

- Виводить підсумкові дані: обчислену суму ряду, точне значення стандартної функції з бібліотеки `<math.h>`, абсолютну похибку та кількість фактично витрачених ітерацій.

3. Скомпілюйте, перевірте на збіжність при різних значеннях ε та зафіксуйте у Git:

```
gcc -Wall -Wextra task2.c -o task2 -lm
git add task2.c
git commit -m "feat(lab2): implement infinite series convergence
calculation"
```

Етап 3. Формування та публікація звітності

1. Оформіть пояснювальну записку до лабораторної роботи відповідно до вимог **Розділу 5** та експортуйте її у файл `Report_Lab02.pdf`.
2. Помістіть згенерований PDF-документ безпосередньо у теку `lab02`.
3. Зафіксуйте звіт у репозиторії та надішліть усі оновлення на віддалений сервер GitHub:

```
git add Report_Lab02.pdf
git commit -m "docs(lab2): add comprehensive engineering PDF report"
git push origin main
```

---

4. ІНДИВІДУАЛЬНІ ВАРІАНТИ ЗАВДАНЬ

Номер варіанта визначається номером у списку академічного журналу групи:

$$\text{Варіант} = (\text{Номер у списку} - 1) \pmod{15} + 1$$

Таблиця 1. Варіанти до Завдання 1 (Кусочно-неперервні функції)

Вар.	Аналітичний вираз кусочної функції $Y(X)$
1	$X \le 0 : Y = -X; \quad 0 < X \le 2 : Y = X^2; \quad X > 2 : Y = 4.0$
2	$X < -1 : Y = \frac{1}{X}; \quad -1 \le X \le 1 : Y = \sqrt{1 - X^2}; \quad X > 1 : Y = X - 1$
3	$X \le 1 : Y = 2X + 1; \quad 1 < X \le 3 : Y = \frac{X}{X-2}; \quad X > 3 : Y = \sqrt{X}$
4	$X < 0 : Y = X^3; \quad 0 \le X \le 3.14159 : Y = \sin(X); \quad X > 3.14159 : Y = -1.0$
5	$X \le -2 : Y = 4.0; \quad -2 < X < 2 : Y = X^2 - 1; \quad X \ge 2 : Y = 2X$
6	$X < 0 : Y = \frac{1}{1+X^2}; \quad 0 \le X \le 2 : Y = 1 - 2X; \quad X > 2 : Y = 3\sqrt{X}$

Вар.	Аналітичний вираз кусочної функції $Y(X)$
7	$X \le 1 : Y = \cos(X); \quad 1 < X \le 4 : Y = \frac{1}{X-1}; \quad X > 4 : Y = 2X - 5$
8	$X < -3 : Y = -3.0; \quad -3 \le X \le 0 : Y = \frac{X}{3}; \quad X > 0 : Y = X^2 + 1$
9	$X \le 0 : Y = 1 - X^2; \quad 0 < X \le 1 : Y = \sqrt{X}; \quad X > 1 : Y = \frac{1}{X}$
10	$X < 2 : Y = 3X - 2; \quad 2 \le X \le 5 : Y = \frac{X^2}{X-1}; \quad X > 5 : Y = 10.0$
11	$X \le -1 : Y = X + 1; \quad -1 < X < 1 : Y = 1 - X^2; \quad X \ge 1 : Y = X - 1$
12	$X < 0 : Y = \frac{2}{X-3}; \quad 0 \le X \le 3 : Y = \sqrt{9 - X^2}; \quad X > 3 : Y = X - 3$
13	$X \le 1 : Y = X^3 - 1; \quad 1 < X \le 2 : Y = \frac{2}{X}; \quad X > 2 : Y = 2.0$
14	$X < -2 : Y = -X - 2; \quad -2 \le X \le 2 : Y = \frac{X}{2}; \quad X > 2 : Y = X^2 - 3$
15	$X \le 0 : Y = 2X; \quad 0 < X \le 3 : Y = \frac{1}{X+1}; \quad X > 3 : Y = \sqrt{X - 3}$

Таблиця 2. Варіанти до Завдання 2 (Збіжність нескінченних чисельних рядів)

Вар.	Функція та її аналітичний розклад у ряд	Область збіжності
1	$e^X = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{X^n}{n!}$	$-\infty < X < \infty$
2	$\sin(X) = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n X^{2n+1}}{(2n+1)!}$	$-\infty < X < \infty$
3	$\cos(X) = 1 - \frac{X^2}{2!} + \frac{X^4}{4!} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n X^{2n}}{(2n)!}$	$-\infty < X < \infty$
4	$\ln(1 + X) = X - \frac{X^2}{2} + \frac{X^3}{3} - \frac{X^4}{4} + \dots = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} X^n}{n}$	$-1 < X \le 1$
5	$\frac{1}{1-X} = 1 + X + X^2 + X^3 + \dots = \sum_{n=0}^{\infty} X^n$	$\|X\| < 1$
6	$\text{sh}(X) = X + \frac{X^3}{3!} + \frac{X^5}{5!} + \dots = \sum_{n=0}^{\infty} \frac{X^{2n+1}}{(2n+1)!}$	$-\infty < X < \infty$
7	$\text{ch}(X) = 1 + \frac{X^2}{2!} + \frac{X^4}{4!} + \dots = \sum_{n=0}^{\infty} \frac{X^{2n}}{(2n)!}$	$-\infty < X < \infty$
8	$\text{arctg}(X) = X - \frac{X^3}{3} + \frac{X^5}{5} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n X^{2n+1}}{2n+1}$	$\|X\| \le 1$
9	$\frac{1}{1+X} = 1 - X + X^2 - X^3 + \dots = \sum_{n=0}^{\infty} (-1)^n X^n$	$\|X\| < 1$
10	$\ln\left(\frac{1+X}{1-X}\right) = 2\left(X + \frac{X^3}{3} + \frac{X^5}{5} + \dots\right) = 2\sum_{n=0}^{\infty} \frac{X^{2n+1}}{2n+1}$	$\|X\| < 1$
11	$\frac{1}{(1-X)^2} = 1 + 2X + 3X^2 + 4X^3 + \dots = \sum_{n=1}^{\infty} n X^{n-1}$	$\|X\| < 1$
12	$\sqrt{1 + X} = 1 + \frac{X}{2} - \frac{X^2}{8} + \frac{X^3}{16} - \dots$ *(біноміальний розклад)*	$\|X\| \le 1$
13	$\frac{1}{\sqrt{1+X}} = 1 - \frac{X}{2} + \frac{3X^2}{8} - \frac{5X^3}{16} + \dots$	$\|X\| < 1$

Вар.	Функція та її аналітичний розклад у ряд	Область збіжності
14	$e^{-X^2} = 1 - \frac{X^2}{1!} + \frac{X^4}{2!} - \frac{X^6}{3!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n X^{2n}}{n!}$	$-\infty < X < \infty$
15	$\cos^2(X) = 1 - \frac{2X^2}{2!} + \frac{8X^4}{4!} - \frac{32X^6}{6!} + \dots$	$-\infty < X < \infty$

---

5. ВИМОГИ ДО ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ (PDF)

Пояснювальна записка є обов'язковим технічним звітом інженера про виконане дослідження. Вона оформлюється згідно з правилами академічної доброчесності та стандартами ЄСДП (Єдина система програмної документації).

Файл звіту повинен мати суворе ім'я `Report_Lab02.pdf` та розміщуватися в каталозі `lab02` репозиторію.

Обов'язкова структура документа:

1. **Титульний аркуш:**
   - Назва закладу вищої освіти, інституту/факультету, кафедри.
   - Назва дисципліни, номер та тема лабораторної роботи.
   - Номер індивідуального варіанта.
   - Відомості про виконавця: студент(ка), курс, академічна група, повне ім'я та прізвище.
   - Відомості про викладача: посада, науковий ступінь, прізвище та ініціали.
   - Місто та рік виконання.
2. **Мета роботи та індивідуальне завдання:**
   - Точне формулювання завдання для Завдання 1 та Завдання 2 із зазначенням математичних виразів вашого варіанта.
3. **Математичний аналіз та рекурентні співвідношення:**
   - Для Завдання 1: аналітичний розрахунок ОДЗ (області допустимих значень) функції, визначення точок розриву.
   - Для Завдання 2: **детальний аналітичний вивід формули переходу** $R(n, X) = \frac{a_n}{a_{n-1}}$. Необхідно розписати скорочення факторіалів та степенів, показавши кінцевий коефіцієнт множення.
4. **Опис алгоритмів та логічних рішень:**
   - Текстове пояснення логіки роботи умовних операторів та циклів.
   - Опис захисних заходів: як саме програма перевіряє вхідні дані та запобігає зацикленню.
5. **Лістинг вихідного коду:**

- Повні, не скорочені лістинги програмних файлів `task1.c` та `task2.c`.
- Код має бути відформатований моноширинним шрифтом (наприклад, Consolas, Courier New або DejaVu Sans Mono) із збереженням інженерних відступів та коментарів.

6. **Результати тестування та верифікація:**
   - Таблиці результатів табулювання із консолі.
   - Зведена таблиця збіжності ряду: значення X, точність ε, отримана сума, еталонне значення з `<math.h>`, відносна похибка та кількість витрачених ітерацій для різних значень ε (наприклад, для $\varepsilon = 10^{-2}, 10^{-4}, 10^{-6}$).
   - Знімки екрана (скріншоти), що демонструють поведінку програм при некоректних вхідних даних (від'ємний крок, вихід за межі області збіжності).
7. **Висновки:**
   - Оцінка швидкості збіжності ряду при віддаленні від точки розкладу.
   - Аналіз ефективності рекурентного обчислення порівняно з прямим обчисленням функцій.
   - Висновок щодо досягнення мети роботи.

---

6. ЗМІСТ РЕПОЗИТОРІЮ ДЛЯ ЗДАЧІ РОБОТИ

До очного захисту допускається робота, завантажена на GitHub у такій структурі каталогів:

```
programming-sem1/
├── .gitignore
├── README.md
├── lab01/
│   ├── pipeline.c
│   ├── memory_audit.c
│   └── task3.c
└── lab02/
    ├── task1.c
    ├── task2.c
    └── Report_Lab02.pdf
```

---

7. КОНТРОЛЬНІ ЗАПИТАННЯ ДЛЯ ОЧНОГО ЗАХИСТУ

1. Як процесор на апаратному рівні реалізує інструкцію розгалуження? Яку роль у цьому відіграють прапорці `ZF` (Zero Flag) та `SF` (Sign Flag) у регістрі стану `EFLAGS`?

2. Що таке проблема «підвішеного else» (Dangling Else) і чому інженерні стандарти безпеки (наприклад, MISRA C) вимагають обов'язкового використання фігурних дужок для кожного блоку розгалуження?
3. Чим відрізняється поведінка операторів `break` та `continue` всередині параметричного циклу `for`? На якій саме стадії циклу передається керування при виклику кожної з цих команд?
4. Що станеться, якщо організувати зворотний відлік за допомогою беззнакового типу даних:
   `unsigned int counter = 10; while (counter >= 0) { counter--; }`? Поясніть фізику явища переповнення розрядної сітки.
5. У чому полягає апаратне явище Branch Misprediction? Чому хаотичні, непередбачувані розгалуження всередині циклу можуть знизити швидкість роботи програми в 5–10 разів?
6. Доведіть, чому рекурентне обчислення наступного члена ряду $a_n = a_{n-1} \cdot R(n, X)$ потребує менше тактових циклів процесора, ніж пряме обчислення через бібліотечні функції `pow()` та факторіал.
7. Що таке «гарячий цикл» (Hot Loop)? Чому розмір тіла циклу, що перевищує обсяг кешу L1 Instruction Cache (32–64 КБ), викликає різке падіння продуктивності системи?
8. За яких умов компілятор оптимізує оператор множинного вибору `switch-case` у таблицю переходів (Jump Table)? Яка часова складність вибору гілки в такій таблиці?