Exercise Activity – Fall 2025 Midterm (10/8/2025)

You can tear the front page, but be sure to write your name/ID on the second page.

First Last Name:_____
ID:_____
Lab Section: _____

- 4 total questions will be given.
- Lowest score dropped: Only your best 3 questions will count toward the grade.
- The last question is graded as all-or-nothing (either full points or zero).
- Maximum total = 110%.
- The pseudocode section is worth up to 10% per question.
- Code comments are worth 5% — just include a few key logic explanations.

Q1. Taylor Series Approximate sin(x) and cos(x) using Taylor series:
sin(x) = x − x^3/3! + x^5/5! − …
cos(x) = 1 − x^2/2! + x^4/4! − …

$$\sin(z) = z - \frac{1}{3!}z^3 + \frac{1}{5!}z^5 - \frac{1}{7!}z^7 + \frac{1}{9!}z^9 \pm \cdots = \sum \frac{(-1)^n}{(2n+1)!}z^{2n+1}$$

$$\cos(z) = 1 - \frac{1}{2!}z^2 + \frac{1}{4!}z^4 - \frac{1}{6!}z^6 + \frac{1}{8!}z^8 - \frac{1}{10!}z^{10} \pm \cdots = \sum \frac{(-1)^n}{(2n)!}z^{2n}$$

Tasks
1) Read x (radians) and error tolerance eps (>0).
2) Compute sin(x) and cos(x) by adding terms until the next term's absolute value < eps.
3) Print the number of terms used for each, the approximations, and compare them with math library values.
4) If |cos(x)| is not too small, print tan(x) ≈ sin(x)/cos(x) and the library tan(x).
Rules
• Use loops and helper functions (e.g., factorial or iterative term update).
• Stop early when precision eps is met.

Test Case 1 (Input):
x = 1.5708 (≈ π/2), eps = 1e−6
Output:
sin ≈ 0.999999 (terms≈6), lib=1.000000
cos ≈ 0.000002 (terms≈5), lib≈0.000000
tan: large (division by tiny cos)

Test Case 2 (Input):
x = 3.1416 (≈ π), eps = 1e−5
Output:
sin ≈ −0.000002 (terms≈7), lib=0.000000
cos ≈ −1.000003 (terms≈8), lib=−1.000000
tan ≈ 0.000002, lib≈0.000000

Q2:. Bitwise Operations – Permissions and Fields
A 16-bit unsigned integer stores three fields:
- Bits 0–3: permissions (bit0=read, bit1=write, bit2=exec, bit3=admin)
- Bits 4–7: group code (0–15)
- Bits 8–15: user code (0–255)
Tasks
1. Extract each field using bitwise operators (&, |, ^, ~, <<, >>).
2. Write a function to set or clear a permission bit k (0–3).
3. Write a function to update the group code to a new 4-bit value v (0–15).
4. Print all fields before and after updates.

Test Cases Case 1:
Input:
value = 0xABCD
set k=1(k is position) → 1 (set write)
set group → 5
Output:
Before: perm=0xD group=0xC user=0xAB
After : perm=0xF group=0x5 user=0xAB

Test Cases Case 2:
Input:
value = 0x000F
clear k=3 (admin)
set group → 0xF
Output:
Before: perm=0xF group=0x0 user=0
After : perm=0x7 group=0xF user=0

Q3 Write a C program to merge two 1D arrays into a third array that includes duplicates, **ordered by the number of duplicates**. If two numbers have the same count, sort them in ascending order.(33.33 points)

Input example:
First array: 7 7 7 6 6 5 4
Second array: 2 2 2 3 3 4 5 5 5 5 6 7

Expected output:
Merged array: [5, 7, 2, 6, 3, 4]

Explanation:
5 appears 5 times
7 appears 4 times
2 appears 3 times
6 appears 3 times
3 appears 2 times
4 appears 1 time

Q4: Write a C program that manages a simple list of student records using arrays.
Each record has three fields: Name, ID, and Section.

Your program should:
1. Read initial student records from a text file (input.txt). Each line has the format:
   Name,ID,Section
2. Process commands from the same file:
   • add Name ID Section → Add a new student.
   • del ID → Delete the student with that ID.
3. After finishing, sort the records by Name (alphabetical order).
4. Print the final list in the format:
5.End programming by typing "end".
   Name, ID, Section

Example Test Case
Input file (input.txt):
Lily,12345,L01
Bob,99123,L02
Carol,55555,L01

del 99123
Expected Output:
Carol,55555,L01
Lily,12345,L01

add Dave 22222 L03
Expected Output:
Carol,55555,L01
Dave,22222,L03
Lily,12345,L01

add Bob 99123 L02
Expected Output:
Bob,99123,L02
Carol,55555,L01
Dave,22222,L03
Lily,12345,L01