Computing Bootcamp: C programming

Final exam

2025.02.07. Fri 14:00 ~ 16:00

[General instructions]

- Do not include any additional header files beyond those already provided.
- Consider edge cases on your own as part of your solution.
- Complete the function in the space provided. Do not modify any other parts of the program.
- Submit the provided skeleton file with the original filename as specified for proper grading.
- Remove all printf statements added for debugging before submitting.
- There are a total of 6 questions. Please refer to the different point values for each question and solve them accordingly.
- You should NOT share your code with other students. Any student found to have a high level of code similarity, as determined by our similarity detection process, may face severe penalties.

1. numOf0s (8 points)

❖ Filename to be submitted: prob1.c

Write a function numOf0s() that takes an integer n as input and counts the number of '0's in its binary representation.

Input

• A single integer 'n' where 1 <= n <= 10^9.

Output

• int num that represents the number of '0's in n's binary representation.

Examples

[Test with prob1.c] ./prob1 3 4 5	
[Output] 0 2 1	

2. Find the Least Common Multiple (8 points)

❖ Filename to be submitted: prob2.c

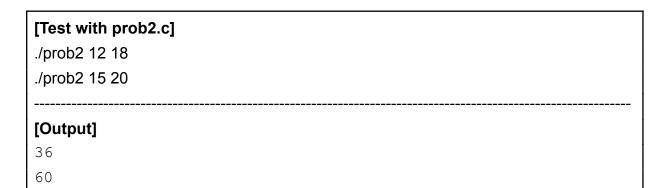
Your task is to implement a function that calculates the **Least Common Multiple** (**LCM**) of two given positive integers.

- 1) Implement the 'gcd' function
 - Use Euclidean Algorithm to compute the GCD.
 - This function will be used inside the 'lcm' function.
- 2) Implement the 'Icm' function
 - The function should take **two integers** as arguments.
 - It should return their Least Common Multiple (LCM).
 - The LCM should be computed using the formula:

$$lcm(a, b) = \frac{|a \times b|}{gcd(a,b)}$$

The input numbers will always be **positive integers** (≥ 1).

Examples



3. Outer Product of Two Vectors (12 points)

Filename to be submitted: prob3.c

Your task is to implement a function that calculates the **Outer Product** of two given vectors.

Outer Product:

$$a \otimes b = ab^{T} = \begin{bmatrix} a_{1} \\ a_{2} \\ \vdots \\ a_{n} \end{bmatrix}_{(n \times 1)} \begin{bmatrix} b_{1} & b_{2} & \dots & b_{m} \end{bmatrix} = \begin{bmatrix} a_{1}b_{1} & \cdots & a_{1}b_{m} \\ \vdots & \ddots & \vdots \\ a_{n}b_{1} & \cdots & a_{n}b_{m} \end{bmatrix}_{(n \times m)}$$

The input will be two random sized arrays $(n, m \ge 1)$.

The function should fill in the **result 2-D array** (**n x m**).

Examples

[Test with prob3.c]

./prob3 n m a1 a2 \dots an b1 b2 \dots bm

./prob3 3 3 1 2 3 4 5 6

[Output]

4 5 6

8 10 12

12 15 18

4. Fill in the structure (12 points)

Filename to be submitted: prob4.c

The provided 'Car' structure and 'set_mileage' function is incomplete. Your task is to analyze the main function and determine what components are required. After that, implement the following:

- 1) Implement the 'Car' structure
 - Define a Car structure with appropriate members based on the information available in the "main" function.
- 2) Implement the 'set_mileage' function
 - The function 'set_mileage' should calculate and store the mileage of a car based on distance traveled and fuel consumed using the formula:

$$mileage = \frac{distance traveled}{fuel consumed}$$

 If the fuel consumed is zero, set mileage to 0 (to avoid division by zero).

You **should not modify the 'main' function**. All the information needed to implement the 'Car' structure and 'set_mileage' function can be obtained from the 'main' function.

Examples

[Test with prob4.c]

./prob4 101 500 25 102 300 20 103 450 30

[Output]

```
Car ID: 101, Mileage: 20.00 km/l
Car ID: 102, Mileage: 15.00 km/l
Car ID: 103, Mileage: 15.00 km/l
```

5. Merge Two Sorted Linked Lists (15 points)

❖ Filename to be submitted: prob5.c

You are provided with a partial implementation of a C program that manages a **singly linked list**. Your task is to complete the implementation of the function 'merge_lists' that merges two already sorted linked lists.

Input

The function takes a pointer to the head of two linked lists as its parameter
 >> merge_lists(struct Node* a, struct Node* b)

Output

• The function should return the head of the new merged linked list.

Examples

[Test with prob5.c]

./prob5 [size of linked list a] [size of linked list b] a1 a2 \dots an b1 b2 \dots bn ./prob5 3 4 1 3 5 2 4 6 8

[Output]

1 2 3 4 5 6 8

6. Find the longest word in a text file (15 points)

❖ Filename to be submitted: prob6.c

You need to implement a function 'findLongestWord()' which reads a given text file and finds the longest word in the file. Words are separated by spaces or newline characters (\n). If multiple words have the same maximum length, the first occurring word should be printed.

OHINT: a string must end with a null character ('\0').

Input

• filename: The name of the text file to be processed

Output

• Print the longest word found in the file on a single line (length <= 1000)

Examples

[input.txt] This is a programming exam	
[Test with prob6.c] ./prob6 input.txt	
[Output] programming	