🐱 GitHub Classroom Setup Practice

This assignment is designed to get you familiar with GitHub Classroom assignment submission and grading. Please watch the video tutorial links on the Canvas course page to understand the steps in detail. Please note that this is not a graded assignment.

Objective

Write a C program that reads two integers and performs basic arithmetic operations.

Problem Description

Your program should:

1. Read two integers from standard input

2. Calculate and output their sum, difference, and product

3. Handle the operations correctly for all integer inputs

Requirements

- Use these GCC compiler flags: `gcc -std=c17 -Wall -Wextra -Werror -pedantic -g -O0`
- Read input using: `scanf()`
- Output results using: `printf()`
- Handle both positive and negative integers
- Follow good programming practices

Input Format

Two integers separated by whitespace on a single line.

Output Format

Three lines containing:

1. The sum of the two integers

2. The difference (first integer minus second integer)

3. The product of the two integers

Examples

Example 1

Input:

```
1  12.5
2  10.3
```

Output:

```
1  Enter two numbers
2  Addition Result = 22.799999
3  Subtraction Result = 2.200000
4  Multiplication Result = 128.750000
5  Division Result = 1.213592
6  Successful execution. Exiting program...
```

Example 2

Input:

```
1  12.5
2  -10.3
```

Output:

```
1  Enter two numbers
2  Addition Result = 2.200000
3  Subtraction Result = 22.799999
4  Multiplication Result = -128.750000
5  Division Result = -1.213592
6  Successful execution. Exiting program...
```

Example 3

Input:

```
1  12.5
2  0
```

Output:

```
1  Enter two numbers
2  Addition Result = 12.500000
3  Subtraction Result = 12.500000
4  Multiplication Result = 0.000000
5  ERROR: divisio-by-zero. Aborting...
```

Submission Instructions

1. Edit the file `skeleton_code.c` with your implementation

2. Test your code locally if possible

3. Commit and push your changes to submit

4. Check the "Actions" tab to see your autograder results

5. You can submit multiple times before the deadline

Grading Breakdown

- **Compilation (10 points)**: Your code must compile without errors
- **Test Cases (90 points)**: Your program must produce correct output for all test cases

Getting Help

- Test your code with the provided examples
- Make sure your output format matches exactly (no extra text)
- Check for common issues: missing semicolons, incorrect printf format
- Ask questions during office hours

Academic Integrity

- Write your own code
- Do not share your solution with other students
- You may discuss the problem approach but not share actual code
- All submissions are automatically checked for similarity