Name: Kiet Chu
Assignment 1

Part 1

- This README summarizes pass@k across two model families and two prompting strategies (each run with 3 samples). Source: `results.txt`.
- Prompts were submitted to ChatGPT and Claude AI through their respective web interfaces. No automated inference code was used - all results were recorded manually by copying responses and evaluating correctness against the provided test cases.
- Prompting Strategies Used
  - Chain-of-Thought (CoT)
  - Self-Planning
- Each strategy generated 3 completions per problem and per model, for a total of 6 samples per model-problem pair.

ProblemID	claude_pass@1	claude_pass@2	claude_pass@3	gpt_pass@1	gpt_pass@2	gpt_pass@3	notes
54	1.000	1.000	1.000	1.000	1.000	1.000	claude: c=6/n=6; gpt: c=6/n=6
151	1.000	1.000	1.000	1.000	1.000	1.000	claude: c=6/n=6; gpt: c=6/n=6
222	1.000	1.000	1.000	1.000	1.000	1.000	claude: c=6/n=6; gpt: c=6/n=6
290	0.000	0.000	0.000	0.000	0.000	0.000	claude: c=0/n=6; gpt: c=0/n=6
375	0.000	0.000	0.000	0.000	0.000	0.000	claude: c=0/n=6; gpt: c=0/n=6
410	1.000	1.000	1.000	1.000	1.000	1.000	claude: c=6/n=6; gpt: c=6/n=6
491	1.000	1.000	1.000	1.000	1.000	1.000	claude: c=6/n=6; gpt: c=6/n=6
596	0.000	0.000	0.000	0.000	0.000	0.000	claude: c=0/n=6; gpt: c=0/n=6
677	0.000	0.000	0.000	0.000	0.000	0.000	claude: c=0/n=6; gpt: c=0/n=6
944	0.000	0.000	0.000	0.000	0.000	0.000	claude: c=0/n=6; gpt: c=0/n=6

- Averages by Model

- claude: pass@1 = 0.500, pass@2 = 0.500, pass@3 = 0.500
- gpt: pass@1 = 0.500, pass@2 = 0.500, pass@3 = 0.500

- Discussion

- Results show consistent performance patterns across both model families, with identical pass@k scores for each problem.
- Both models achieved perfect scores (1.000) on 5 out of 10 problems, while completely failing (0.000) on the remaining 5 problems.
- No performance differences were observed between Chain-of-Thought and Self-Planning strategies, indicating that both approaches were equally effective (or ineffective) for the given problem set.
- Different model families give slightly different answers, but they stay consistent across multiple prompts.
- ChatGPT had more trouble understanding the prompts correctly, sometimes producing responses without proper Python code.
- Both models can misunderstand what the prompts are asking for, leading to wrong answers.

Part 2

This section details the debugging and iterative improvement process for identified failure cases from Part 1.

- Summary of Debugging Efforts

- Changes Made: The primary change involved adding tests directly to the prompt.
- What Worked: Including tests with different cases in the prompt proved effective in improving model performance.
- What Didn't Work: Experimenting with different prompting strategies alone, or adding tests of only similar cases to the prompt, did not yield significant improvements.
- Model Behavior: Both ChatGPT and Claude AI initially failed to correctly understand the problem statements. However, both models successfully generated correct solutions after being provided with relevant test cases within the prompt.

Part 3

- Proposed Strategy: Combine chain-of-thought reasoning with test-guided self-debugging, where the LLM first reasons through the problem step-by-step, then uses failing test cases to iteratively debug its own output.
- Workflow

1. Chain-of-Thought Generation: Model reasons step-by-step privately, then outputs clean code
2. Test-Guided Walkthrough: Run generated code against unit tests, capture failures
3. Self-Debugging Pass: Re-prompt with original problem, generated code, and test failures to fix issues
4. Verification: Re-run tests on revised code, keep best-performing version

- Key Innovation

Unlike baseline strategies that stop after single generation, CoT-SD adds a structured second pass grounded in test feedback, turning passive reasoning into active debugging. This approach is generic and works for any programming problem with known input-output tests.