CS-UH 2012: Software Engineering

**Assignment 3**

**Assignment Deadline**: Check Brightspace and course schedule.

**Assignment Type**: Individual Effort and Individual Submission. **While part of this assignment relates to the application being developed in the group project, you must complete this assignment individually**.

**Submission Method**: Your submission for this assignment should be in PDF format. You will submit your work to Brightspace. Ensure that your document is well-structured, with each section clearly addressing its respective task and providing necessary details as outlined in the assignment instructions.

**Assignment Evaluation and Weight:** The assignment will be graded out of 100 points and is worth 7% of your final course grade.

**Assignment Overview**: This assignment aims to provide hands-on experience with software testing techniques, enabling students to design, implement, and evaluate test cases to ensure software quality and reliability.

**Part I (50 points). This part is related to your project and is expected to inspire and support you in the tasks that are expected for Project Deliverable 4.**

Employing black box testing methodologies, generate a minimum of 15 test cases for the five chosen use cases within your project. Incorporate various black box testing techniques such as Boundary Value Analysis, Equivalence Partitioning, Truth Table, and Pair-Wise testing. For each test case, adhere to the provided template:

**Test Case Title:**

**Use Case Tested by the Test Case:**

**Technique Used to generate test case with a proof (e.g.,** Boundary Value Analysis, Equivalence Partitioning, Truth Table, and Pair-Wise testing)

**Pre-condition:**

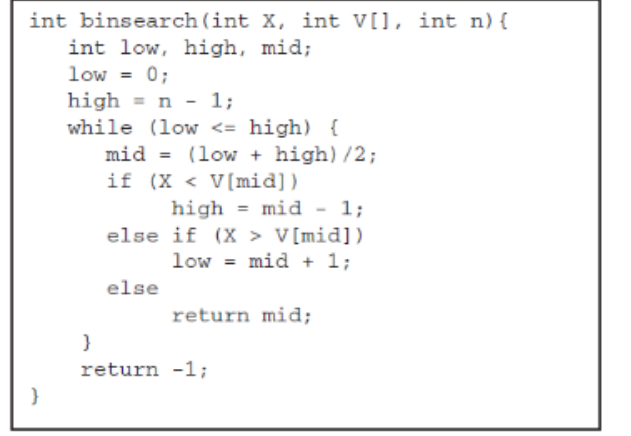
**Input:**

**Steps to Execute:**

**Expected Results:**

**Part II. Answer the following two questions:**

**Question 1 (40 points)**: Consider the following code:



1. [15 points] Draw a Control Flow Graph for the above code. For easier traceability for us when marking, please use the line numbers in the code to label nodes in your graph.
2. [5 points] Calculate the cyclomatic complexity of the code.
3. [15 points] Provide a test suite (i.e., a set of test cases) that would satisfy 100% statement coverage of your CFG. Your test suite should only contain the minimum number of test cases needed to achieve the coverage. **For each test case, provide concrete values of target, numArray, length, and the expected return value.**
4. [5 marks] List any additional test cases **you need to add to your test suite from “c” to achieve 100% branch coverage.**

**Question 2 (10 points)**: Identify a data flow anomaly (*dd, ur, du*) in the code given in Figure. Explain the anomaly

A white screen with black text

AI-generated content may be incorrect.