**Câu 1**: Trình bày các bước nhằm kiểm thử một đơn vị chương trình theo phương pháp kiểm thử dòng điều khiển với một độ đo kiểm thử cho trước

1. **Xác định đơn vị chương trình cần kiểm thử**

* Đơn vị chương trình có thể là một hàm, một module, hoặc một lớp.

1. **Biểu diễn dòng điều khiển dưới dạng đồ thị**

* Trích xuất **Control Flow Graph (CFG)** từ mã nguồn:
  + Mỗi **nút** (node) biểu diễn một khối lệnh tuần tự (basic block).
  + Mỗi **cạnh** (edge) biểu diễn sự chuyển điều khiển (theo điều kiện, vòng lặp, nhảy…).

1. **Chọn độ đo kiểm thử** 
   * **C1:** đảm bảo tất cả các câu lệnh được thực thi ít nhất một lần.
   * **C2** đảm bảo mỗi nhánh điều kiện (true/false) được thực hiện ít nhất một lần.
   * **C3:** đảm bảo mọi đường đi có thể xảy ra đều được kiểm thử.
2. **Xác định tập đường đi kiểm thử theo độ đo đã chọn**

* Từ CFG, liệt kê các đường đi thỏa mãn độ đo.
* Ví dụ:
  + Với **Branch Coverage**, ta cần đảm bảo mỗi điều kiện được đi cả hướng đúng và sai.
  + Với **Path Coverage**, phải liệt kê tất cả các đường đi khả dĩ (nếu khả thi).

1. **Thiết kế ca kiểm thử (Test Cases)**

* Gán giá trị đầu vào để kích hoạt từng đường đi đã chọn.
* Đảm bảo mỗi ca kiểm thử có **đầu vào** (input), **hành động** (execution), và **kết quả mong đợi** (expected output).

1. **Thực thi kiểm thử**

* Chạy đơn vị chương trình với các ca kiểm thử đã thiết kế.
* Ghi lại kết quả thực tế.

1. **So sánh và đánh giá**

* So sánh kết quả thực tế với kết quả mong đợi để xác định **pass/fail**.
* Đánh giá độ bao phủ (coverage achieved) bằng công cụ hoặc thủ công để xem đã đạt mức độ đo kiểm thử yêu cầu chưa.

**Câu 2:**

A white text with black text

AI-generated content may be incorrect.

Giải:

* Đồ thị dòng điều khiển

A screenshot of a computer

AI-generated content may be incorrect.

* Để đạt được độ phủ C1 ta có các đường đi và các đường kiểm thử sau:

P1: 1 – 2 – 3(F) – 5 – 3(T) – 4

Test1: (a [] = [1, 2], n = 2).

* Để đạt được độ phủ C2 ta có các đường đi và các đường kiểm thử sau (TH này giống độ phủ C1):

P1: 1 – 2 – 3(F) – 5 – 3(T) – 4

Test1: (a [] = [1, 2], n = 2).

* Do ta không biết được n là bao nhiêu nên ta sẽ kiểm thử cho vòng lặp for với 4 điều kiện đầu (n = 0, 1, 2, k với 2 < k < n)

Các ca kiểm thử như sau:

Vòng for chạy 0 lần: n = 0; a = []

Vòng for chạy 1 lần: n = 1; a = [1]

Vòng for chạy 2 lần: n = 2; a = [1, 2]

Vòng for chạy 5 lần: n = 0; a = [1, 2, 3, 4, 5]

**Câu 3:**

A white paper with black text

AI-generated content may be incorrect.

* Đồ thị dòng điều khiển

A screenshot of a computer

AI-generated content may be incorrect.

* Để đạt được độ phủ C1 ta có các đường đi và các đường kiểm thử sau:

P1: 1 – 2 – 3(T) – 4 – exit.

* test1: score = -1

P2: 1 – 2 – 3(F) – 5(T) – 6 – 14 – exit.

* test2: score = 9

P3: 1 – 2 – 3(F) – 5(F) – 7(T) – 8 – 14 – exit.

* test3: score = 8

P4: 1 – 2 – 3(F) – 5(F) – 7(F) – 9(T) – 10 – 14 – exit.

* test4: score = 7

P5: 1 – 2 – 3(F) – 5(F) – 7(F) – 9(F) – 11(T) - 12 – 14 – exit.

* test5: score = 6

P6: 1 – 2 – 3(F) – 5(F) – 7(F) – 9(F) – 11(F) - 13 – 14 – exit.

* test6: score = 4
* Các đường đi và các ca kiểm thử ở câu trên vừa có thể thỏa mã độ phủ C2

**Câu 4:**

A white background with black text

AI-generated content may be incorrect.

* Đồ thị dòng điểu khiển

A diagram of a system

AI-generated content may be incorrect.

* Các đường đi và các ca kiểm thử ứng với độ đo C2

P1: 1 – 2(case 65) – 3 – 7

* Test1: x = 65

P2: 1 – 2(case 66) – 4 – 7

* Test1: x = 66

P3: 1 – 2(case 67) – 5 – 7

* Test1: x = 67

P4: 1 – 2(default) – 6 – 7

* Test1: x = 0

**Câu 5**: Kiểm thử chương trình của bạn với độ đo kiểm thử C2

**Bài toán giả định**: Một rạp chiếu phim muốn thu hút khách hàng nhân dịp một bộ phim bom tấn ra rạp đã ra chương trình khuyến mãi cho khách hàng là các phần quà tặng kèm khi mua vé. Cụ thể:

|  |  |  |
| --- | --- | --- |
| Số vé | Số lượng bỏng ngô/nước | Quà tặng |
|  |  | Stickers |
|  | Vở |
|  |  | Vở + Bỏng ngô + Nước |
|  | Poster + Chữ ký đạo diễn |

**Yêu cầu**: Báo cáo phân tích, thiết kế các ca kiểm thử, và kiểm thử chương trình của bạn với độ phủ C2.

* **Đồ thị dòng điều khiển**:
* A screenshot of a computer

  AI-generated content may be incorrect.
* **Xây dựng đường đi để việc kiểm thử đạt mức C2**:
  1. (P1) 1 – 2(T) – 3 – exit

Expected Output: “Invalid Input”.

Testcase: (-1, -1).

* 1. (P2) 1 – 2(F) – 4(T) – 5(T) – 6 – exit

Expected Output: “Stickers”.

Testcase: (3, 5).

* 1. (P3) 1 – 2(F) – 4(T) – 5(F) – 8(T) – 9 – exit

Expected Output: “Notebook”.

Testcase: (3, 9).

* 1. (P4) 1 – 2(F) – 4(T) – 5(F) – 8(F) – 10 – exit

Expected Output: “No Present”.

Testcase: (3, 3).

* 1. (P5) 1 – 2(F) – 4(T) – 5(F) – 7(F) - 10 – exit

Expected Output: “No Present”.

Testcase: (2, 10).

* 1. (P6) 1 – 2(F) – 4(F) – 7(T) – 11(T) - 12 – exit

Expected Output: “Notebook+Popcorn+Water”.

Testcase: (6, 5).

* 1. (P7) 1 – 2(F) – 4(F) – 7(T) – 11(F) – 13(F) -10 – exit

Expected Output: “No Present”.

Testcase: (6, 4).

* 1. (P8) 1 – 2(F) – 4(F) – 7(T) – 11(F) – 13(T) - 14 – exit

Expected Output: “Poster + Signature”.

Testcase: (6, 9).

* **Bảng kết quả lần 1:**



**(8/8)**

[Link Github](https://github.com/quangnhat09z/TestingHomework)