**BÀI TẬP TUẦN**

**Yêu cầu:** Chọn và trả lời 5 câu hỏi cuối 2 chương CFG và DFG

1. **Kiểm thử dòng điều khiển CFG**
2. Trình bày các độ đo kiểm thử cho kiểm thử dòng điều khiển

**Trả lời**

Các độ đo cơ bản gồm:

* Độ đo C1: Mỗi câu lệnh trong chương trình được kiểm tra ít nhất 1 lần.
* Độ đo C2: Các điểm quyết định trong đồ thị dòng điều khiển của chương trình được thực hiện ít nhất 1 lần ở cả 2 nhánh đúng/sai
* Độ đó C3: Các điều kiện con nằm trong diều kiện phức tạp ở các điểm quyết định trong đồ thị dòng điều khiển cần được kiểm thử ít nhất 1 lần tại cả 2 nhánh đúng/sai.

1. Trình bày các bước kiểm thử một đơn vị chương trình theo phương pháp kiểm thử dòng điều khiển

**Trả lời**

* Xây dựng đồ thị dòng điều khiển của chương trình
* Xác định tiêu chuẩn kiểm thử sẽ sử dụng
* Tìm các đường đi trên đồ thị thỏa mãn tiêu chuẩn
* Sinh các ca kiểm thử từ các đường đi
* Thực hiện kiểm thử

1. Cho hàm:

char Grade( int score ){

int res; (1)

if(scrore<0 || scrore>10) (2)

return ‘I’; (3)

if(score>=9) (4)

res = ‘A’; (5)

else

if(score>=8) (6)

res = ‘B’; (7)

else

if(score>=6.5) (8)

res = ‘C’; (9)

else

if(score>=5) (10)

res = ‘D’; (11)

else

res = ‘F’; (12)

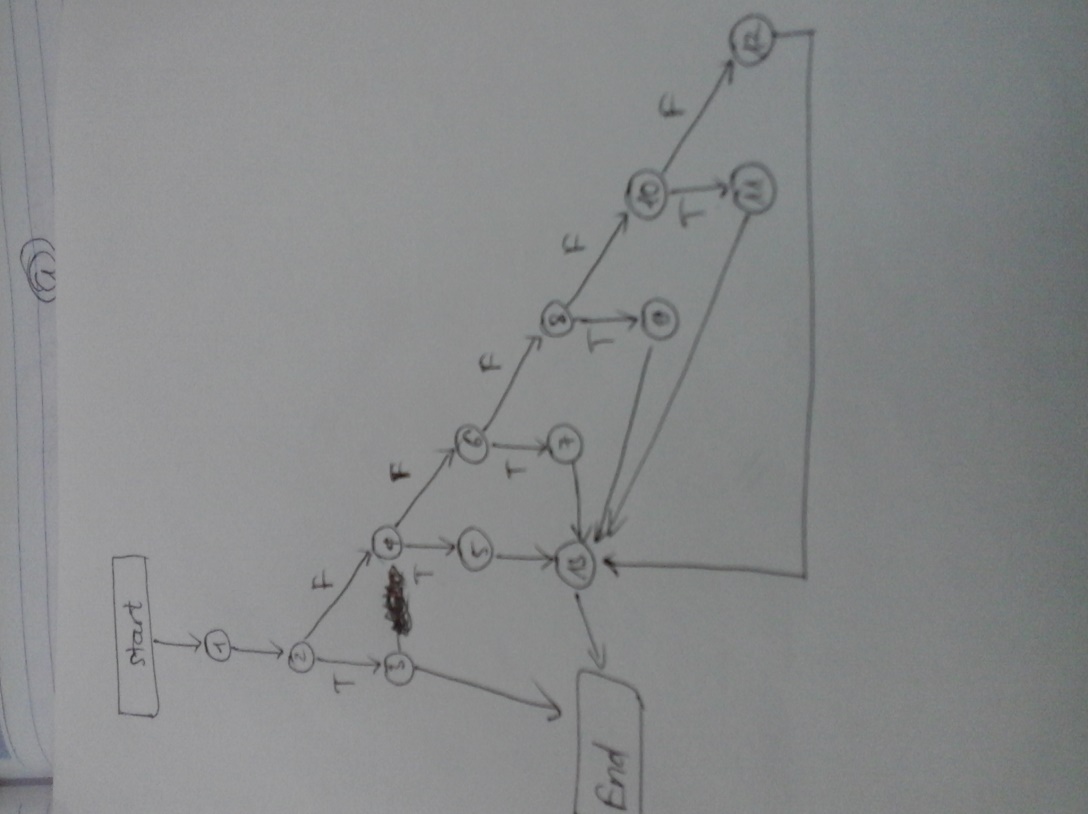
return res; (13)

}

1. Xây dựng đồ thị dòng điều khiển cho hàm Grade ứng với độ đo C1 và C2
2. Sinh các đường đi và ca kiểm thử cho độ đo C1
3. Sinh các đường đi và ca kiểm thử cho độ đo C2

**Trả lời**

1. Đồ thị dòng điều khiển:



1. Đường đi và ca kiểm thử theo tiêu chuẩn C1:

|  |  |  |
| --- | --- | --- |
| STT | Đường đi | Ca kiểm thử |
| 1 | 1, 2, 3 | score = 14 |
| 2 | 1, 2, 4, 5, 13 | score = 10 |
| 3 | 1, 2, 4, 6, 7, 13 | score = 8 |
| 4 | 1, 2, 4, 6, 8, 9, 13 | score = 6.5 |
| 5 | 1, 2, 4, 6, 8, 10, 11, 13 | score = 6 |
| 6 | 1, 2, 4, 6, 8, 10, 12, 13 | score = 3 |

1. Đường đi và ca kiểm thử theo tiêu chuẩn C2:

Trường hợp này, các đường đi và ca kiểm thử giống tiêu chuẩn C1

1. Cho chương trình:

int LaSoNguyenTo (int n){

int i = 2; (1)

do{

if ( (n%i)==0 ) (2)

return 0; (3)

i++; (4)

} while (i<=n/2); (5)

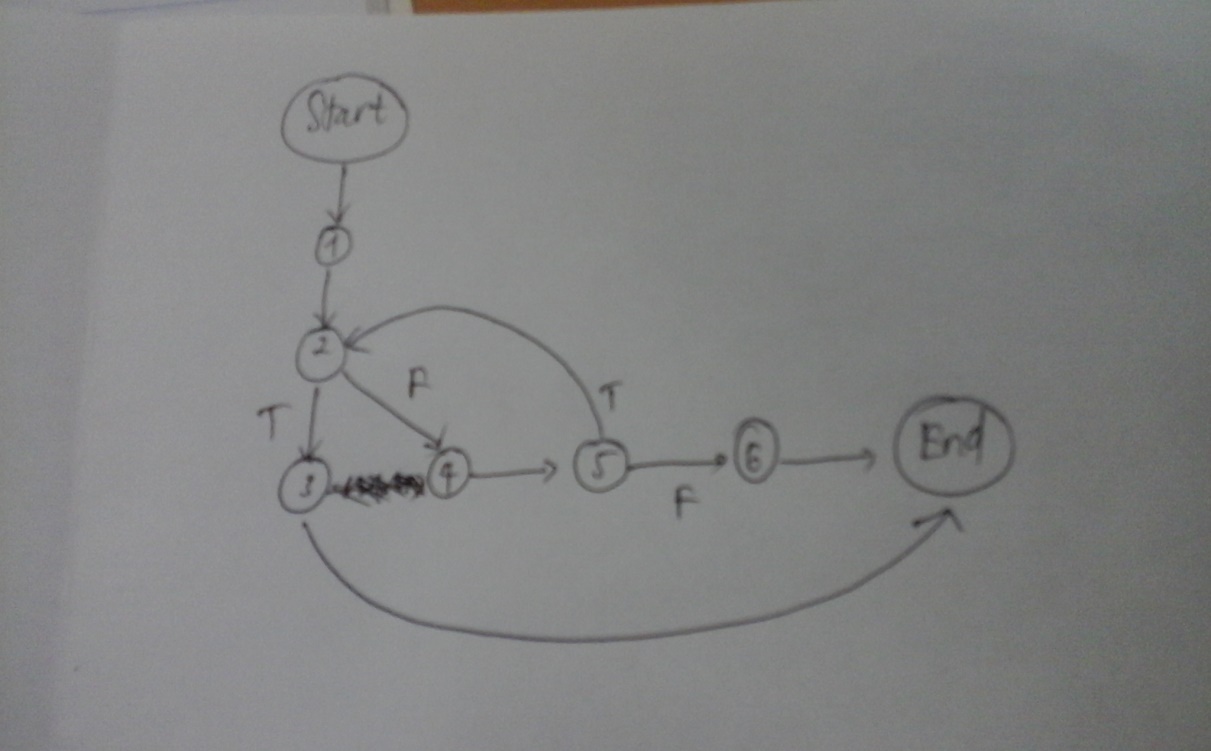
return 1; (6)

}

1. Xây dựng đồ thị dòng điều khiển cho hàm Grade ứng với độ đo C1 và C2
2. Sinh các đường đi và ca kiểm thử cho độ đo C1
3. Sinh các đường đi và ca kiểm thử cho độ đo C2

**Trả lời**

1. Đồ thị dòng điều khiển:



1. Đường đi và ca kiểm thử theo tiêu chuẩn C1:

|  |  |  |
| --- | --- | --- |
| STT | Đường đi | Ca kiểm thử |
| 1 | 1, 2, 3 | n = 10 |
| 2 | 1, 2, 4, 5, 6 | n = 5 |
| 3 | 1, 2, 4, 5, 2, 4, 5, 6 | n = 7 |
| 4 | 1, 2, 4, 5, 2, 3 | Không tồn tại n thỏa mãn |

1. Đường đi và ca kiểm thử theo tiêu chuẩn C2:

Trường hợp này, các đường đi và ca kiểm thử giống tiêu chuẩn C1

1. **Kiểm thử dòng dữ liệu DFG**
2. Mô tả ba loại vấn đề phổ biến về dòng dữ liệu. Với mỗi loại vấn đề, hãy lấy  
   ví dụ minh họa.

**Trả lời**

Ba vấn đề phổ biến gồm:

* Gán giá trị rồi gán tiếp giá trị

Ví dụ: int a = 10;

a = 20;

* Chưa gán giá trị nhưng đã sử dụng

Ví dụ: int a, b, c;

a = 5;

c = a + b; // biến b chưa được gán giá trị

return c;

* Đã được khai báo và gán giá trị nhưng không được sử dụng

Ví dụ: int Sum( int a, int b){

int c = 0, k = 2; // biến k không được sử dụng

c = a + b;

return c;

}

1. Trình bày các bước trong quy trình kiểm thử dòng dữ liệu động

**Trả lời**

* Xây dựng đồ thị dòng dữ liệu cho chương trình
* Chọn tiêu chuẩn kiểm thử sẽ sử dụng
* Xác định các đường đi trong đồ thị theo tiêu chuẩn đã chọn
* Giải biểu thức điều kiện từ tập các đường đi để chọn ra các ca kiểm thử
* Thực hiện kiểm thử

1. Cho hàm calFactorial viết bằng ngôn ngữ C như hình.
2. Hãy liệt kê các câu lệnh ứng với các khái niệm *def*, *c−use*, và *p−use*  
   ứng với các biến được sử dụng trong hàm này.
3. Hãy vẽ đồ thị dòng dữ liệu của hàm này

int calFactorial (int n){

int result = 1; (1)

int i = 1; (2)

while (i<=n){ (3)

result = result \*i; (4)

i++; (5)

}

return result; (6)

}

**Trả lời**

1. - Lệnh ứng với ‘def’: (1), (2), (5)

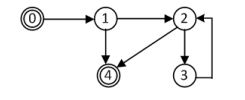
* Lệnh ứng với ‘c-use’: (4)
* Lệnh ứng với ‘p-use’: (3)

1. Đồ thị dòng dữ liệu:

1 2 3 4 5

6

1. Cho đồ thị dòng dữ liệu:



1. Hãy xác định tất cả các *Complete*-*path* từ đồ thị này.
2. Các đường đi (0 - 1 - 2 - 3), (1 - 2 - 3 - 2 - 4), (3 - 2 - 4), (2 - 3 - 2) và (0 - 1 - 4) có phải là các simple-paths không? Giải thích.
3. Các đường đi (2 - 3 - 2), (2 - 3) và (3 - 2 - 4) có phải là các *loop*-*freepaths* không? Giải thích.

**Trả lời**

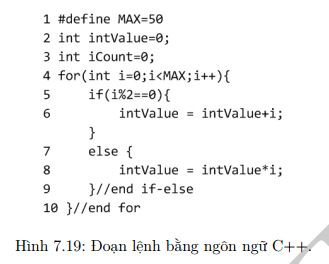
1. Các *Complete*-*path* từ đồ thị là: (0-1-2-4), (0-1-4), (0-1-2-3-2-4), (0-1-2-3-2-3-2-3…-2-4)
2. Simple-paths là các đường đi mà các đỉnh chỉ xuất hiện đúng 1 lần trừ đỉnh đầu và đỉnh cuối.

Do vậy, các đường đi là simple-paths là: (0 - 1 - 2 - 3), (3 - 2 - 4), (2 - 3 - 2) và (0 - 1 - 4)

1. *loop*-*freepaths* là đường đi mà các đỉnh chỉ xuất hiện đúng 1 lần.

Do vậy, các đường đi là *loop*-*freepaths* là: (2 - 3) và (3 - 2 - 4)

1. Cho đoạn lệnh:



1. Biến *i* trong vòng lặp *for* có ảnh hưởng đến giá trị của biến *intValue*  
   không? Tại sao?
2. Hãy xác định các các câu lệnh ảnh hưởng đến giá trị của biến *intValue*  
   tại các câu lệnh 6 và 8.

**Trả lời**

1. Biến i trong vòng for làm ảnh hưởng đến giá trị của biến intValue vì giá trị của intValue thay đổi theo giá trị của biến i.

Nếu biến i là chẵn, intValue = intValue+i

Nếu i lẻ, intValue = intValue\*i

1. Cả 2 câu lệnh (6) và (8) đều làm ảnh hưởng đến giá trị biến intValue.