

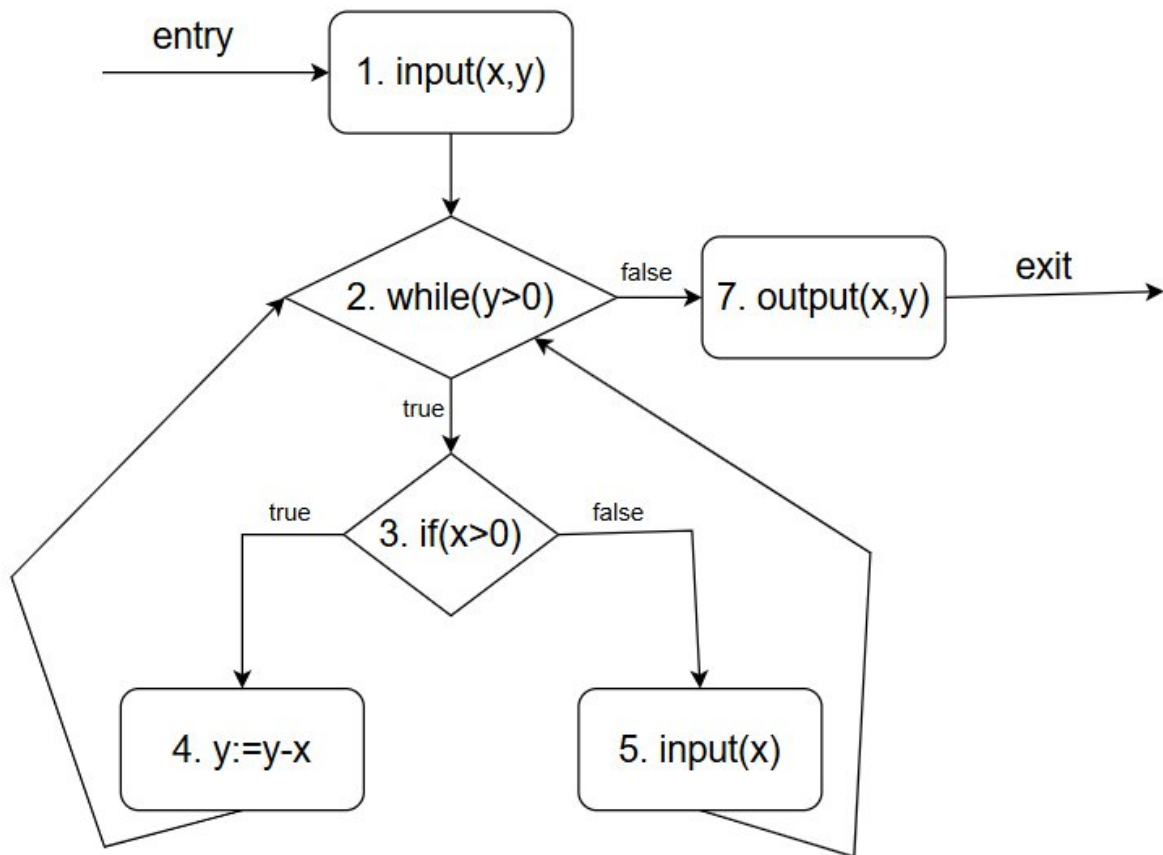
Bài 1:

Quy trình tổng quát của kiểm thử dòng dữ liệu động:

- Vẽ đồ thị luồng điều khiển (CFG)
- Lựa chọn tiêu chí kiểm thử luồng dữ liệu
- Xác định các đường đi trên CFG thỏa mãn tiêu chí kiểm thử đã chọn
- Sinh các ca kiểm thử tương ứng

Bài 2:

- Đồ thị CFG



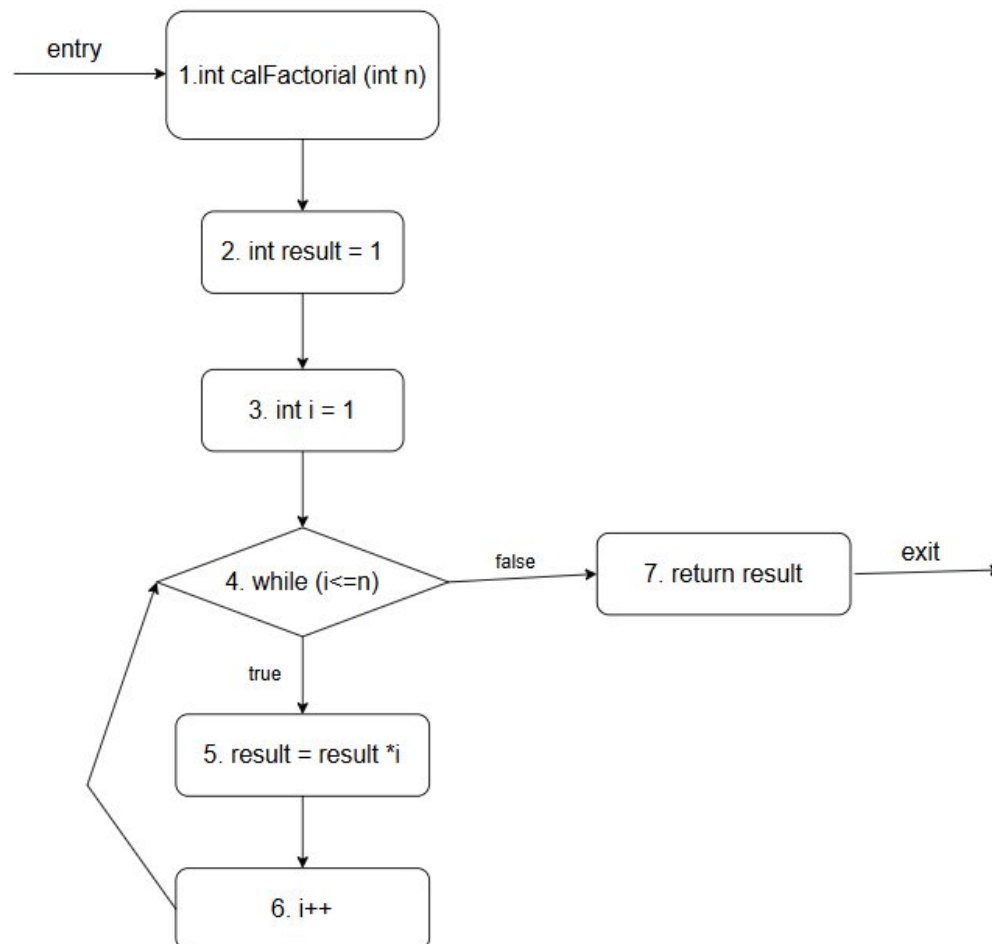
- Xác định du-pairs
 - + Biến x: $\text{def}(x) = \{1,5\}$, $\text{p-use}(x) = \{3\}$, $\text{c-use}(x) = \{4,7\}$
 - + Biến y: $\text{def}(y) = \{1,4\}$, $\text{p-use}(y) = \{2\}$, $\text{c-use}(y) = \{4,7\}$
- Sinh đường đi và các ca kiểm thử với từng Test Cases

| Variable | Du-pairs | Complete Path | Test cases | |
|----------|----------|-------------------------------------|------------|-----------------|
| | | | Input | Expected output |
| X | (1,3(T)) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=3, y=2 | (3, -1) |
| | (1,3(F)) | 1 -> 2(T) -> 3(F) -> 5 -> 2(F) -> 7 | x=0, y=2 | (2, -1) |

| | | | | |
|---|----------|--|----------|---------|
| | (1,4) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=3, y=2 | (3, -1) |
| | (1,7) | 1 -> 2(F) -> 7 | x=5, y=0 | (5, 0) |
| | (5,3(T)) | 1 -> 2(T) -> 3(F) -> 5 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=0, y=2 | (3, -1) |
| | (5,3(F)) | 1 -> 2(T) -> 3(F) -> 5 -> 2(T) -> 3(F) -> 5 -> 2(F) -> 7 | x=0, y=2 | (2, -1) |
| | (5,4) | 1 -> 2(T) -> 3(F) -> 5 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=0, y=2 | (3, -1) |
| | (5,7) | 1 -> 2(T) -> 3(F) -> 5 -> 2(F) -> 7 | x=0, y=2 | (3, -1) |
| Y | (1,2(T)) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=1, y=2 | (1,0) |
| | (1,2(F)) | 1 -> 2(F) -> 7 | x=5, y=0 | (5, 0) |
| | (1,4) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=3, y=2 | (3,-1) |
| | (1,7) | 1 -> 2(F) -> 7 | x=5, y=0 | (5, 0) |
| | (4,2(T)) | 1 -> 2(T) -> 3(T) -> 4 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=2, y=5 | (2, -1) |
| | (4,2(F)) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=3, y=2 | (3, -1) |
| | (4,7) | 1 -> 2(T) -> 3(T) -> 4 -> 2(F) -> 7 | x=3, y=2 | (3, -1) |

Bài 3:

- Đồ thị CFG:



- Các câu lệnh ứng với các khái niệm def, c-use, p-use ứng với các biến sử dụng trong hàm là:

| Dòng | Câu lệnh | def | c-use | p-use |
|------|--------------------------|--------|-----------|-------|
| 1 | int calFactorial (int n) | n | | |
| 2 | int result = 1; | result | | |
| 3 | int i = 1; | i | | |
| 4 | while (i <= n) | | | i, n |
| 5 | result = result * i; | result | result, i | |
| 6 | i++; | i | i | |
| 7 | return result; | | result | |

Bài 4:

- Các def-clear-path ứng với x và y:

+ Với biến x:

0 -> 1 -> (đến p-use tại cạnh 1→2)
 0 -> 1 -> (đến p-use tại cạnh 1→3)
 0 -> 1 -> 2 -> 4 -> (đến các p-use tại 4→5 và 4→6, tiếp tục tới 5 hoặc 6)
 0 -> 1 -> 2 -> 4 -> 5
 0 -> 1 -> 2 -> 4 -> 5 -> 6
 0 -> 1 -> 2 -> 4 -> 6
 3 -> 4 -> (đến p-use tại 4→5 và 4→6)
 3 -> 4 -> 5 -> (đến c-use(x) ở nút 5)
 3 -> 4 -> 6

+ Với biến y:

0 -> 1 -> (các p-use ở cạnh 1→2 và 1→3)
 0 -> 1 -> 3 -> 4 -> 6
 0 -> 1 -> 3 -> 4 -> 5
 2 -> 4 -> (đến p-use tại 4→5 và 4→6)
 2 -> 4 -> 6
 2 -> 4 -> 5 -> 6
 5 -> 6

- Các du-paths ứng với x và y

+ Với biến x:

0 -> 1 -> (p-use trên cạnh 1→2).
 0 -> 1 -> (p-use trên cạnh 1→3).
 0 -> 1 -> 2 -> 4 -> (p-use ở 4→5).
 0 -> 1 -> 2 -> 4 -> (p-use ở 4→6).
 0 -> 1 -> 2 -> 4 -> 5
 3 -> 4 -> (p-use ở 4→5).
 3 -> 4 -> (p-use ở 4→6).

3 -> 4 -> 5.

+ Với biến y:

0 -> 1-> (p-use trên cạnh 1->2)
 0 -> 1-> (p-use trên cạnh 1->3)
 0 -> 1 -> 3 -> 4 -> (p-use trên cạnh 4->5 và 4->6)
 0 -> 1 -> 3 -> 4 -> 6.
 2 -> 4 -> (p-use ở 4->5).
 2 -> 4 -> (p-use ở 4->6)
 2 -> 4 -> 6.
 5 -> 6.

- Các All-p-uses / Some-c-uses và All-c-uses/Some-p-uses là:

| Variable | All-p-uses/Some-c-uses | All-c-uses/Some-p-uses |
|----------|--|--|
| X | (0->1), (0->1->2->4), (3->4), (0->1->3) | (0->1->3), (0->1), (3->4) |
| y | (0->1), (2->4), (2->4->6) | (0->1->2->4->6), (2->4->6), (5->6), (0->1), (2->4) |

- Khả năng đi trên đường đi 0 - 1 - 3 - 4 - 5 - 6:

Trên đường đi 0 - 1 - 3 - 4 - 5 - 6, ta có thứ tự sự kiện như sau:

- + Tại nút 0: x, y được định nghĩa (giá trị khởi tạo).
- + Tại cạnh (1,3) kiểm tra $x + y = 4$, điều này ràng buộc **giá trị khởi tạo** của x và y .
- + Ở nút 3: có một lệnh **Def(x)** (gán lại x); sau đó (hoặc trước tùy cách trình bày) các bước sau dùng giá trị mới của x .
- + Ở cạnh (4,5) kiểm tra $x^2 + y^2 > 17$; ở đây x là giá trị **sau khi** được định nghĩa ở nút 3, còn y vẫn là giá trị khởi tạo (vì trên đường này ta không đi qua nút 2 hay nút 5 trước đó để gán lại y).

=> Có thể thỏa mãn **cả hai** ràng buộc đồng thời. Ví dụ cụ thể:

+ Chọn giá trị khởi tạo $x = 4, y = 0 \Rightarrow x + y = 4$ thỏa (cạnh (1,3)).

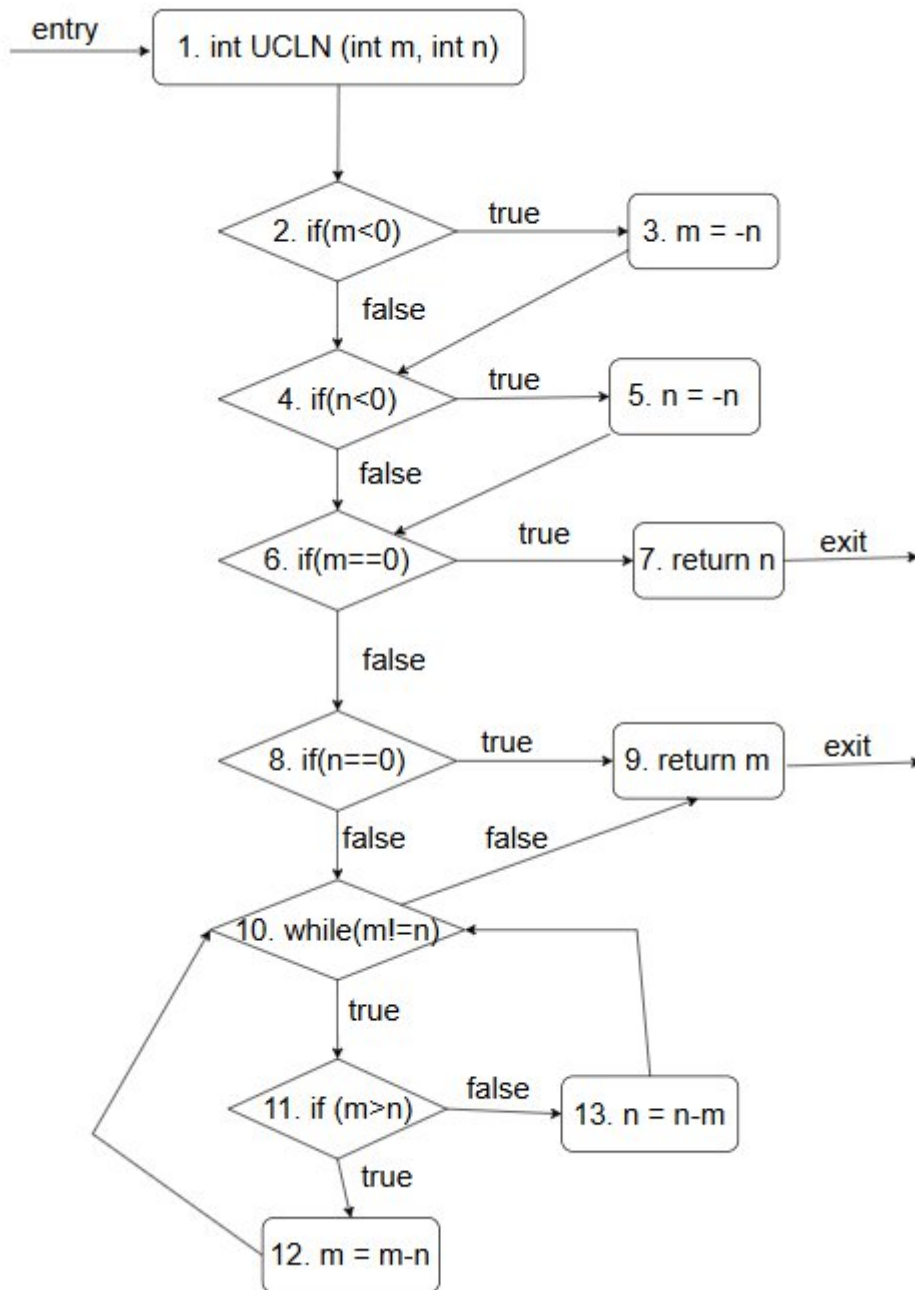
+ Tại nút 3 gán lại $x := 5$. Khi đến cạnh (4,5) ta có $x^2 + y^2 = 5^2 + 0^2 = 25 > 17$.

≡ Vậy đường đi 0 - 1 - 3 - 4 - 5 - 6 là **khả thi** (có thể thực hiện được) — đưa ra được một tập giá trị và một gán tại nút 3 thỏa hai điều kiện đã cho.

- Ở đỉnh 3 biến x vừa được định nghĩa vừa được sử dụng nhưng không có mối quan hệ def-use vì **use ở nút 3 dùng giá trị cũ của x (trước khi def tại 3 thực hiện)** nên def tại nút 3 không phải là nguồn giá trị cho use đó → không hình thành mối quan hệ def-use giữa def(3) và use(3). (Mối liên hệ def-use của use ở 3 nếu có sẽ tới một def trước đó mà không bị ghi đè giữa chặng đường.)

Bài 5:

- Đồ thị CFG:



- Kiểm thử với độ đo C2:

| Testcase | Path | Input | Expected output |
|----------|------|-------|-----------------|
|----------|------|-------|-----------------|

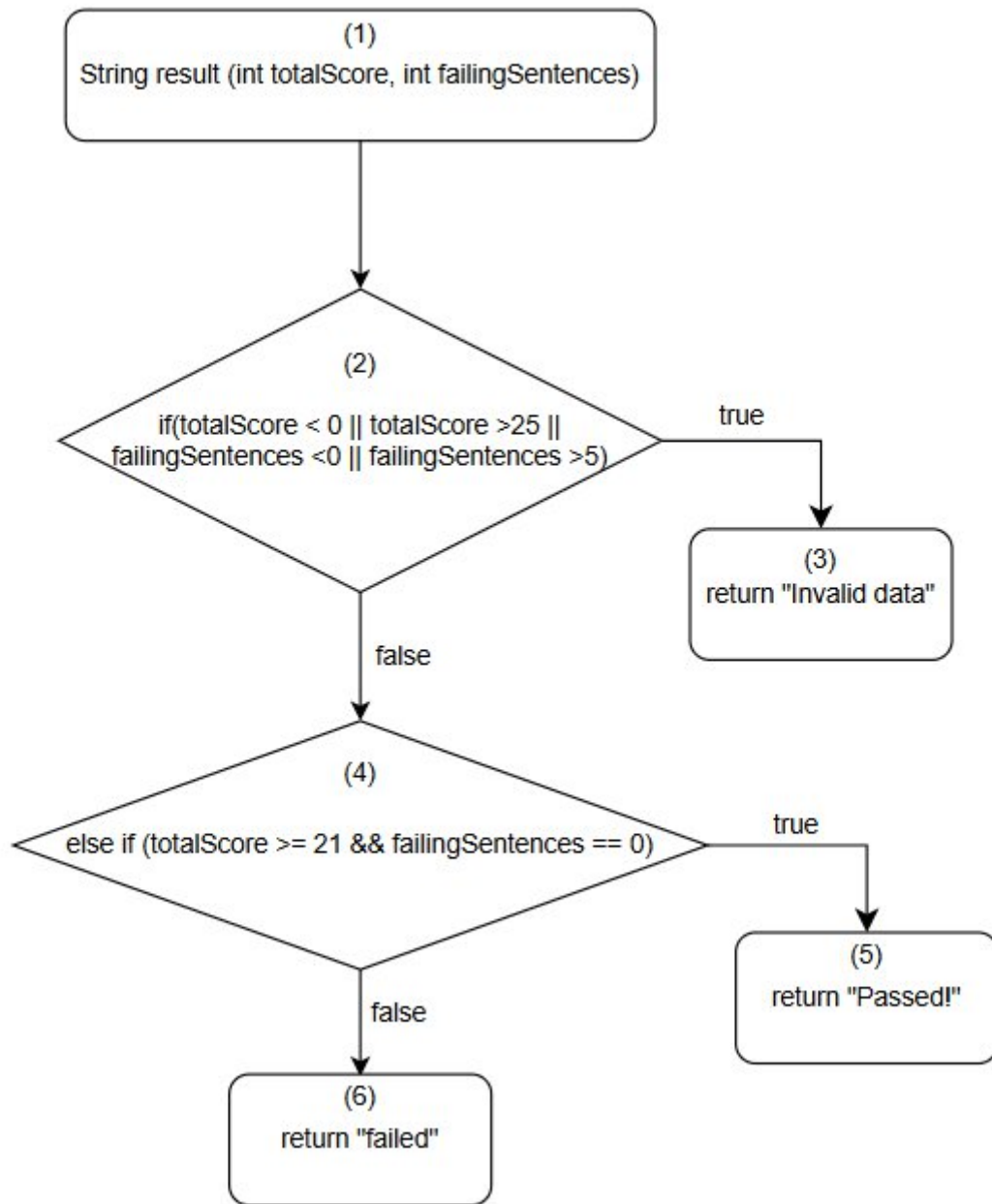
| | | | |
|---|--|----------------|---|
| 1 | 1 -> 2(false) -> 4(false) -> 6(true) -> 7 (exit) | m = 0, n = 5 | 5 |
| 2 | 1 -> 2(false) -> 4(false) -> 6(false) -> 8(true) -> 9 (exit) | m = 7, n = 0 | 7 |
| 3 | 1 -> 2(false) -> 4(false) -> 6(false) -> 8(false) -> 10 (false) -> 9 | m = 5, n = 5 | 5 |
| 4 | 1 -> 2(false) -> 4(false) -> 6(false) -> 8(false) -> -> 10(true) -> 11(true) -> 12 -> 10(false) -> 9 | m = 9, n = 6 | 3 |
| 5 | 1 -> 2(false) -> 4(false) -> 6(false) -> 8(false) -> -> -> 10(true) -> 11(false) -> 13 -> 10(false) -> 9 | m = 6, n = 9 | 3 |
| 6 | 1 -> 2(true) -> 3 -> 4(true) -> 5 -> 6(true) -> 7 (exit) | m = -4, n = -6 | 2 |
| 7 | 1 -> 2(true) -> 3 -> 4(false) -> 6(false) -> 8(true) -> 9 | m = -3, n = 0 | 3 |

- Kiểm thử với độ đo all-def coverage:

| Testcase | Path | Input | Expected output |
|----------|---|---------------|-----------------|
| 1 | 1 -> 2 (false) -> 4 (false) -> 6 (true) -> 7(exit) | m = 0, n = 5 | 5 |
| 2 | 1 -> 2 (false) -> 4 (false) -> 6 (false) -> 8 (true) -> 9(exit) | m = 5, n = 0 | 5 |
| 3 | 1 -> 2 (true) -> 3 -> 4 (false) -> 6 (false) -> 8 (false) -> 10 (false) -> 9 | m = -3, n = 3 | 3 |
| 4 | 1 -> 2 (false) -> 4 (true) -> 5 -> 6 (false) -> 8 (false) -> 10 (false) -> 9 | m = 4, n = -4 | 4 |
| 5 | 1 -> 2(false) -> 4(false) -> 6(false) -> 8(false) -> 10(true) -> 11(true) -> 12 -> 10(true) -> 11(true) -> 12 -> 10(true) -> 11(false) -> 13 -> 10(true) -> 11(false) -> 13 -> 10(false) -> 9 | m = 7, n = 3 | 1 |

Bài 6:

- Mã nguồn: <https://github.com/dongVu1105/TestingExcercise.git>
(Mã nguồn ở folder Week5)
- Đồ thị CFG:



- Kiểm thử theo độ phủ All-uses:

| Variable | Du-pairs | Complete Path | Test cases | | Actually output |
|------------------|----------|-------------------------|------------|-----------------|-----------------|
| | | | Input | Expected output | |
| totalScore | (1,2(T)) | 1 -> 2(T) -> 3 | (-1, -1) | "Invalid data" | "Invalid data" |
| | (1,2(F)) | 1 -> 2(F) -> 4 (T) -> 5 | (22, 0) | "Passed!" | "Passed!" |
| | (1,4(T)) | 1 -> 2(F) -> 4 (T) -> 5 | (22, 0) | "Passed!" | "Passed!" |
| | (1,4(F)) | 1 -> 2(F) -> 4 (F) -> 6 | (18, 1) | "failed" | "failed" |
| failingSentences | (1,2(T)) | 1 -> 2(T) -> 3 | (-1, -1) | "Invalid data" | "Invalid data" |
| | (1,2(F)) | 1 -> 2(F) -> 4 (T) -> 5 | (22, 0) | "Passed!" | "Passed!" |
| | (1,4(T)) | 1 -> 2(F) -> 4 (T) -> 5 | (22, 0) | "Passed!" | "Passed!" |
| | (1,4(F)) | 1 -> 2(F) -> 4 (F) -> 6 | (18, 1) | "failed" | "failed" |