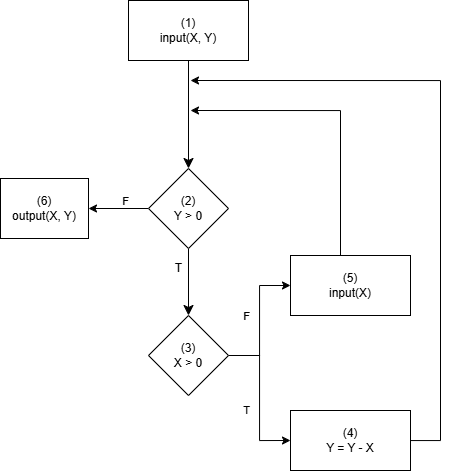
**Bài 1:**

Các bước trong quy trình kiểm thử dòng dữ liệu động là:

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

**Bài 2:**

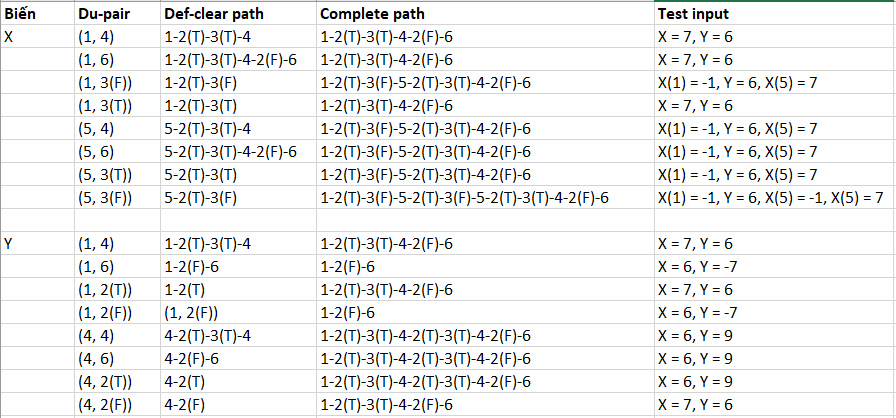
1. Đồ thị CFG



2, 3:

**Biến X:** def(X) = {1, 5}, c-use(X) = {4, 6}, p-use(X) = {3}

**Biến Y:** def(Y) = {1, 4}, c-use(Y) = {4, 6}, p-use(Y) = {2}



**Bài 3:**

Biến n:

* Def: int calFactorial (int n)
* C-use: không có
* P-use: while (i <= n)

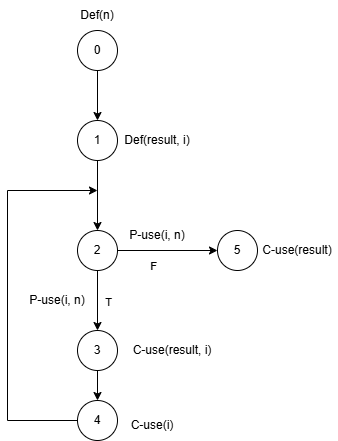
Biến result:

* Def:
  + int result = 1
  + result = result \* i
* C-use:
  + result = result \* i
  + return result
* P-use: không có

Biến i:

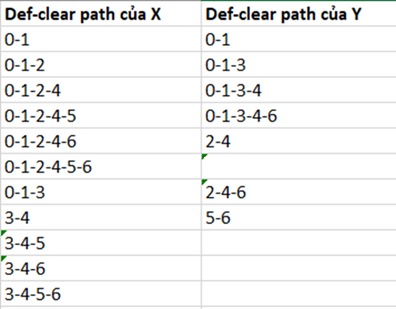
* Def:
  + int i = 1
  + i++
* C-use:
  + result = result \* i
  + i++
* P-use: while (i <= n)

Đồ thị dòng dữ liệu:



**Bài 4:**

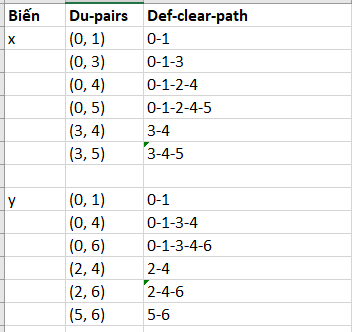
1.



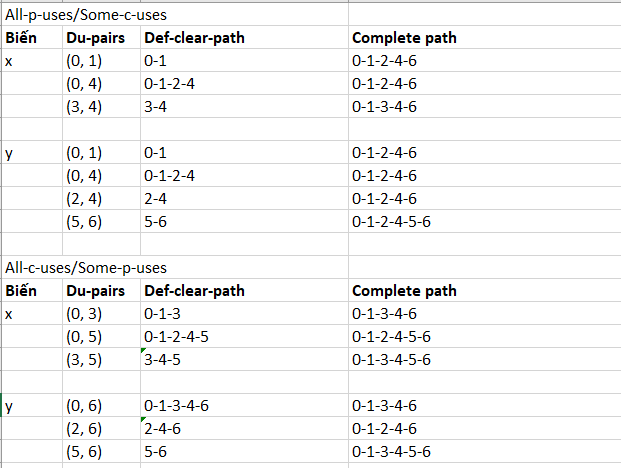
2.

Biến X: Def = {0, 3}; C-uses = {3, 5}; P-uses = {1, 4}

Biến Y: Def = {0, 2, 5}; C-uses = {6}; P-uses = {1, 4}



3.



4.

Giả sử x = 1, y = 3

Đi qua điều kiện x + y == 4 true ở cạnh (1, 2)

Thực thi def(x) c-use(x) ở node 3, giả sử x = x + 2.

Khi đó, x = 3

Khi đó x^2 + y^2 = 3^2 + 3^2 = 18 > 17

=> Đi qua được cạnh (4, 5)

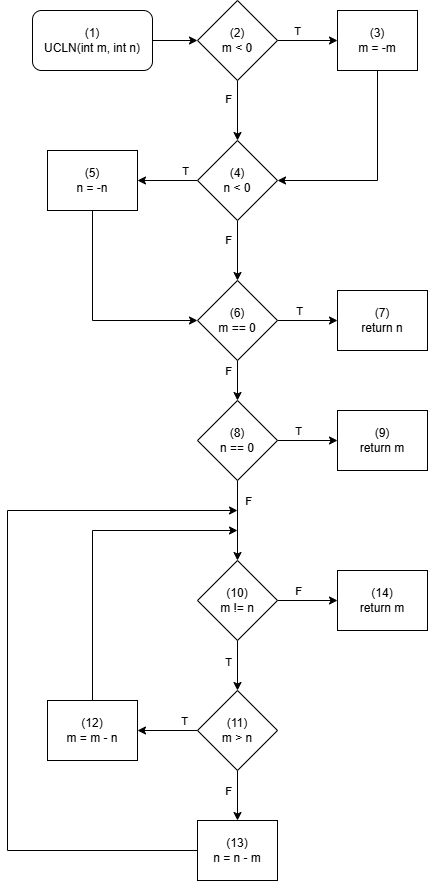
=> Tiếp tục đi được 4 🡪 5 🡪 6.

5.

Không tồn tại quan hệ def-use vì trong cùng 1 đỉnh, use(x) sẽ được sử dụng trước def(x)

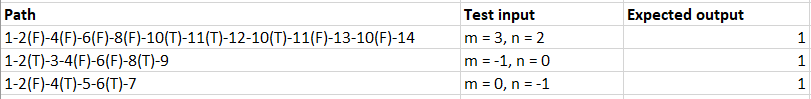
**Bài 5:**

1.



2.

Các đường đi & ca kiểm thử với độ phủ C2



3.

Biến **m**: Def = {1, 3, 12}; C-uses = {3, 9, 12, 13, 14}; P-uses = {2, 6, 10, 11}

Biến **n**: Def = {1, 5, 13}; C-uses = {5, 7, 12, 13}; P-uses = {4, 8, 10, 11}

Các đường đi & ca kiểm thử với độ đo all-defs

