

# Bài 1

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

**Bước 1:** Xây dựng biểu đồ luồng điều khiển (Control Flow Graph – CFG)

**Bước 2:** Xác định các điểm định nghĩa và sử dụng biến (def, c-use, p-use)

**Bước 3:** Xác định các cặp Def–Use (DU pairs)

**Bước 4:** Dựa vào du-pairs để tìm ra def-clear-path, complete-path

**Bước 5:** Xây dựng các ca kiểm thử bao phủ dòng dữ liệu

**Bước 6:** Thực thi kiểm thử dòng dữ liệu động

**Bước 7:** Phân tích kết quả và sửa lỗi

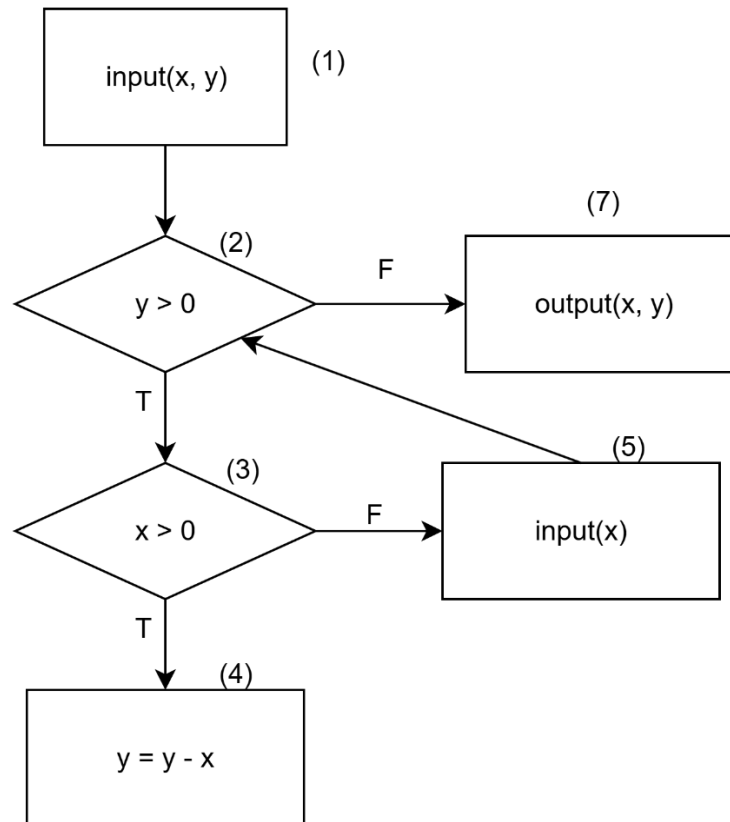
## Bài 2:

```
1. input(X,Y)
2. while (Y>0) {
3.   if (X>0)
4.     Y := Y-X
5.   else
6.     input(X)
7. }
8. output(X,Y)
```

Cho đoạn mã nguồn sau, hãy:

1. Vẽ đồ thị dòng điều khiển (CFG)
2. Xác định các du-pairs cho biến X và Y
3. Sinh đường đi và các ca kiểm thử với độ đo all-use

**1. Đồ thị dòng điều khiển:**



## 2. Xác định các cặp du-pairs:

Ta có:

$\text{def}(x) = \{1, 5\}$

$\text{c-use}(x) = \{4, 7\}$

$\text{p-use}(x) = \{3\}$

Nên: Các du-pairs với biến x là:

$(1, 3(T)), (1, 3(F)), (1, 4), (1, 7), (5, 3(T)), (5, 3(F)), (5, 4), (5, 7)$ .

Lại có:

$\text{def}(y) = \{1, 4\}$

$\text{c-use}(y) = \{4, 7\}$

$\text{p-use}(y) = \{2\}$

Nên: Các du-pairs với biến y là:

$(1, 2(T)), (1, 2(F)), (1, 4), (1, 7), (4, 2(T)), (4, 2(F)), (4, 4), (4, 7)$ .

### 3. Sinh đường đi và các ca kiểm thử:

Variable	du-pair	def-clear-path	complete path	testcase
x	(1, 3(T))	1, 2(T), 3(T)	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)
	(1, 3(F))	1, 2(T), 3(F)	1, 2, 3(F), 5, 2(T), 3(T), 4, 2(F), 7	(-1, 3), 2
	(1, 4)	1, 2(T), 3(T), 4	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)
	(1, 7)	1, 2(F), 7	1, 2(F), 7	(-2, -2)
	(5, 3(T))	5, 2(T), 3(T)	1, 2, 3(F), 5, 2(T), 3(T), 4, 2(F), 7	(-1, 3), 2
	(5, 3(F))	5, 2(T), 3(F)	1, 2, 3(F), 5, 2(T), 3(F), 5, 2(T), 3(T), 4, 2(F), 7	(-1, 1), 10
	(5, 4)	5, 2(T), 3(T), 4	1, 2, 3(F), 5, 2(T), 3(T), 4, 2(F), 7	(-1, 3), 2
	(5, 7)	5, 2(T), 3(T), 4, 2(F), 7	1, 2, 3(F), 5, 2(T), 3(F), 5, 2(T), 3(T), 4, 2(F), 7	(-1, 1), 10
y	(1, 2(T))	1, 2(T)	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)
	(1, 2(F))	1, 2(F)	1, 2(F), 7	(-2, -2)
	(1, 4)	1, 2(T), 3(T), 4	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)
	(1, 7)	1, 2(F), 7	1, 2(F), 7	(-2, -2)
	(4, 2(T))	4, 2(T)	1, 2(T), 3(T), 4, 2(T), 3(T), 4, 2(F), 7	(2, 5)
	(4, 2(F))	4, 2(F)	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)
	(4, 4)	4, 2(T), 3(T), 4	1, 2(T), 3(T), 4, 2(T), 3(T), 4, 2(F), 7	(2, 5)
	(4, 7)	4, 2(F), 7	1, 2(T), 3(T), 4, 2(F), 7	(3, 2)

## Bài 3

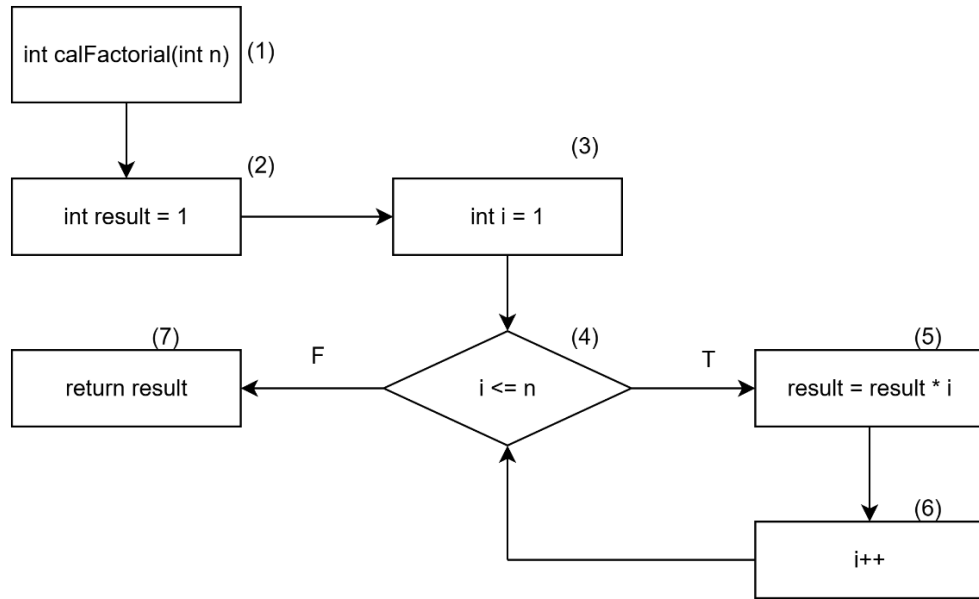
7. Cho hàm `calFactorial` viết bằng ngôn ngữ C như Đoạn mã 7.7.

- 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.
- Hãy vẽ đồ thị dòng dữ liệu của hàm này.

Đoạn mã 7.7: Mã nguồn C của hàm `calFactorial`

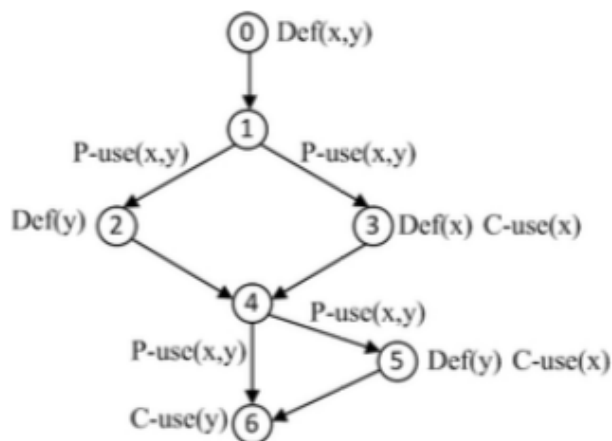
```
1. int calFactorial (int n){  
2.     int result = 1;  
3.     int i=1;  
4.     while (i <= n){  
5.         result = result *i;  
6.         i++;  
7.     }//end while  
    return result;  
}//the end
```

- Các câu lệnh *def*, *c-use*, *p-use* ứng với các biến:  
    `def(n) = {1};`                      `p-use(n) = {4}`  
    `def(result) = {2, 5};`              `c-use(result) = {5, 7}`  
    `def(i) = {3, 6};`              `c-use(i) = {5, 6};`              `p-use(i) = {4}`
- Đồ thị dòng điều khiển



Bài 4:

10. Cho đồ thị dòng dữ liệu như hình 7.11.



Hình 7.11: Một ví dụ về đồ thị dòng dữ liệu và việc sử dụng các biến.

- Hãy xác định tất cả các *Def-clear-path* của các biến  $x$  và  $y$ .
- Hãy xác định tất cả các *du-paths* của các biến  $x$  và  $y$ .
- Hãy xác định tất cả các *All-p-uses/Some-c-uses* và *All-c-uses/Some-p-uses* (dựa vào các chuẩn của kiểm thử dòng dữ liệu).
- Biểu thức của các  $p-use(x, y)$  tại cạnh (1,3) và (4,5) lần lượt là  $x + y = 4$  và  $x^2 + y^2 > 17$ . Đường đi (0 - 1 - 3 - 4 - 5 - 6) có thực thi được không? Giải thích.
- Tại sao tại đỉnh 3 biến  $x$  được định nghĩa và sử dụng nhưng không tồn tại mối quan hệ *def-use*?

1) Các *def-clear-path* ứng với 2 biến  $x$  và  $y$ :

x	y
0, 1	0, 1
0, 1, 2	0, 1, 3
0, 1, 2, 4	0, 1, 2, 3, 4
0, 1, 2, 4, 5	0, 1, 2, 3, 4, 6
0, 1, 2, 4, 5, 6	2, 4
0, 1, 2, 5, 6	2, 4, 6
3, 4	5, 6
3, 4, 5	
3, 4, 5, 6	

2)

Các du-pairs của biến x: (0, 1), (0, 4), (0, 5), (3, 4), (3, 5)

Các du-pairs của biến y: (0, 1), (0, 4), (0, 6), (2, 4), (2, 6), (5, 6)

3)

**\* All p-use/some c-use**

Variable	du-pair	def-clear-path	complete path
x	(0, 1F)	0, 1F	0, 1F, 2, 4F, 6
	(0, 1T)	0, 1T	0, 1T, 3, 4F, 6
	(0, 4F)	0, 1F, 2, 4F	0, 1F, 2, 4F, 6
	(0, 4T)	0, 1F, 2, 4T	0, 1F, 2, 4T, 5, 6
	(3, 4T)	3, 4T	0, 1T, 3, 4T, 5, 6
	(3, 4F)	3, 4F	0, 1T, 3, 4F, 6
y	(0, 1F)	0, 1F	0, 1F, 2, 4F, 6
	(0, 1T)	0, 1T	0, 1T, 3, 4F, 6
	(0, 4F)	0, 1F, 2, 4F	0, 1F, 2, 4F, 6
	(0, 4T)	0, 1F, 2, 4T	0, 1F, 2, 4T, 5, 6
	(2, 4T)	2, 4T	0, 1F, 2, 4T, 5, 6
	(2, 4F)	2, 4F	0, 1F, 2, 4F, 6
	(5, 6)	5, 6	0, 1F, 2, 4T, 5, 6

**\* All c-use/some p-use**

Variable	du-pair	def-clear-path	complete path
x	(0, 3)	0, 1T, 3	0, 1T, 3, 4F, 6
	(0, 5)	0, 1F, 2, 4T, 5	0, 1F, 2, 4T, 5, 6
	(3, 5)	3, 4T, 5	0, 1T, 3, 4T, 5, 6
y	(0, 6)	0, 1T, 3, 4F, 6	0, 1T, 3, 4F, 6
	(2, 6)	2, 4T, 5, 6	0, 1F, 2, 4T, 5, 6
	(5, 6)	5, 6	0, 1F, 2, 4T, 5, 6

4) Để đường đi 0 – 1 – 3 – 4 – 5 – 6 được thực hiện thì 2 điều kiện tại cạnh 1-3 và 4-5 đều là True.

$$\text{Hay } \begin{cases} x + y = 4 \\ x^2 + y^2 > 17 \end{cases}$$

Hệ phương trình này có nghiệm với mọi x, y. Nên 0 – 1 – 3 – 4 – 5 – 6 có thể được thực thi.

5) Mọi quan hệ def-use xảy ra khi:

- + Tại  $d$  là  $\text{def}(x)$
- + Tại  $u$  là  $\text{use}(x)$
- + Trên đường  $(d, u)$  không có lệnh  $\text{def}(x)$

Như vậy tại (3) tồn tại 2 lệnh  $\text{def}(x)$  và  $\text{use}(x)$  nên không có đường đi nào từ  $\text{def}$  đến  $\text{use}$  cả vì  $\text{use}$  xảy ra ngay tại chỗ.



## Bài 5

Cho đoạn mã nguồn như hình bên,

1. Xây dựng CFG cho hàm UCLN với đồ thị C2
2. Sinh đường đi và các ca kiểm thử với độ đo C2
3. Sinh đường đi và các ca kiểm thử với độ đo all-def coverage

Đoạn mã 6.4: Mã nguồn của hàm UCLN

```
int UCLN(int m, int n){  
    if (m < 0) m = -m;  
    if (n < 0) n = -n;  
    if (m == 0) return n;  
    if (n == 0) return m;  
    while (m != n) {  
        if(m > n)  
            m = m - n;  
        else  
            n = n - m;  
    }//end while  
    return m;  
}
```

1. Đồ thị luồng điều khiển:

def(m) = {1, 3, 13}

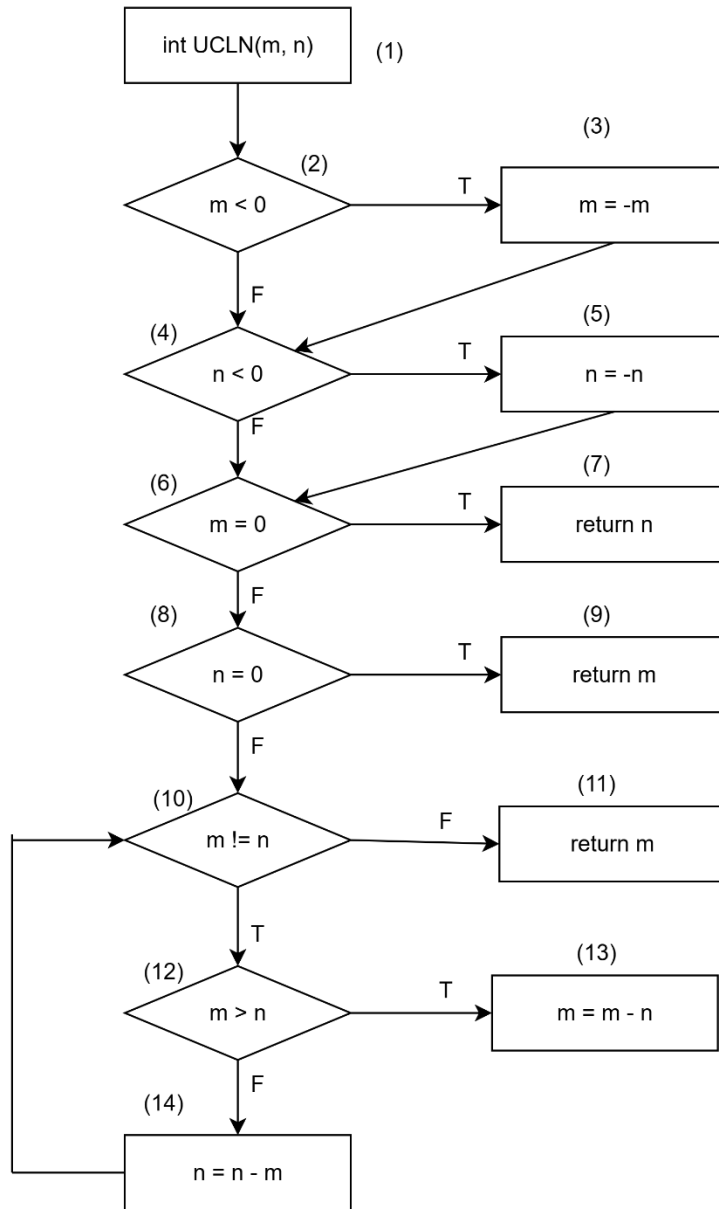
p-use(m) = {2, 3, 6, 10, 12}

c-use(m) = {9, 11, 13, 14}

def(n) = {1, 5, 14}

p-use(n) = {4, 5, 8, 10, 12}

c-use(n) = {7, 13, 14}



## 2. Các đường đi và các ca kiểm thử với độ đo C2:

P1: 1 – 2F – 4F – 6T – 7 - exit

⇒ test1: (0, 2)

P2: 1 – 2F – 4F – 6F – 8T – 9 - exit

⇒ test2: (2, 0)

P3: 1 – 2T – 3– 4T – 6F – 8F – 10T – 12T – 13 – 10F – 11 – exit.

⇒ test3: (-24, -12)

P4: 1 – 2T – 3– 4T – 6F – 8F – 10T – 12F – 14 – 10F – 11 – exit.

⇒ test4: score = (-12, -24)

### 3. Các đường đi và ca kiểm thử độ phủ all-def

Variable	du-pair	def-clear-path	complete path	testcase
x	(1, 6T)	1, 2F, 4F, 6T	1, 2F, 4F, 6T, 7	(0, 3)
	(3, 9)	3, 4F, 6F, 8T, 9	1, 2T, 3, 4F, 6F, 8T, 9	(-3, 0)
	(13, 11)	13, 10F, 11	1, 2T, 3, 4T, 6F, 8F, 10T, 12T, 13, 10F, 11	(-24, -12)
y	(1, 7)	1, 2F, 4F, 6T, 7	1, 2F, 4F, 6T, 7	(0, 3)
	(5, 7)	5, 6T, 7	1, 2F, 4T, 5, 6T, 7	(0, -3)
	(14, 10)	14, 10F	1, 2T, 3, 4T, 6F, 8F, 10T, 12F, 14, 10F, 11	(-12, -24)

## Bài 6

### • Kiểm thử chương trình của bạn với độ phủ all-uses

**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 bóng ngô/nước	Quà tặng
$3 \leq x \leq 5$	$5 \leq y \leq 8$	Stickers
	$y > 8$	Vở
$x > 5$	$5 \leq y \leq 8$	Vở + Bóng ngô + Nước
	$y > 8$	Poster + Chữ ký đạo diễn

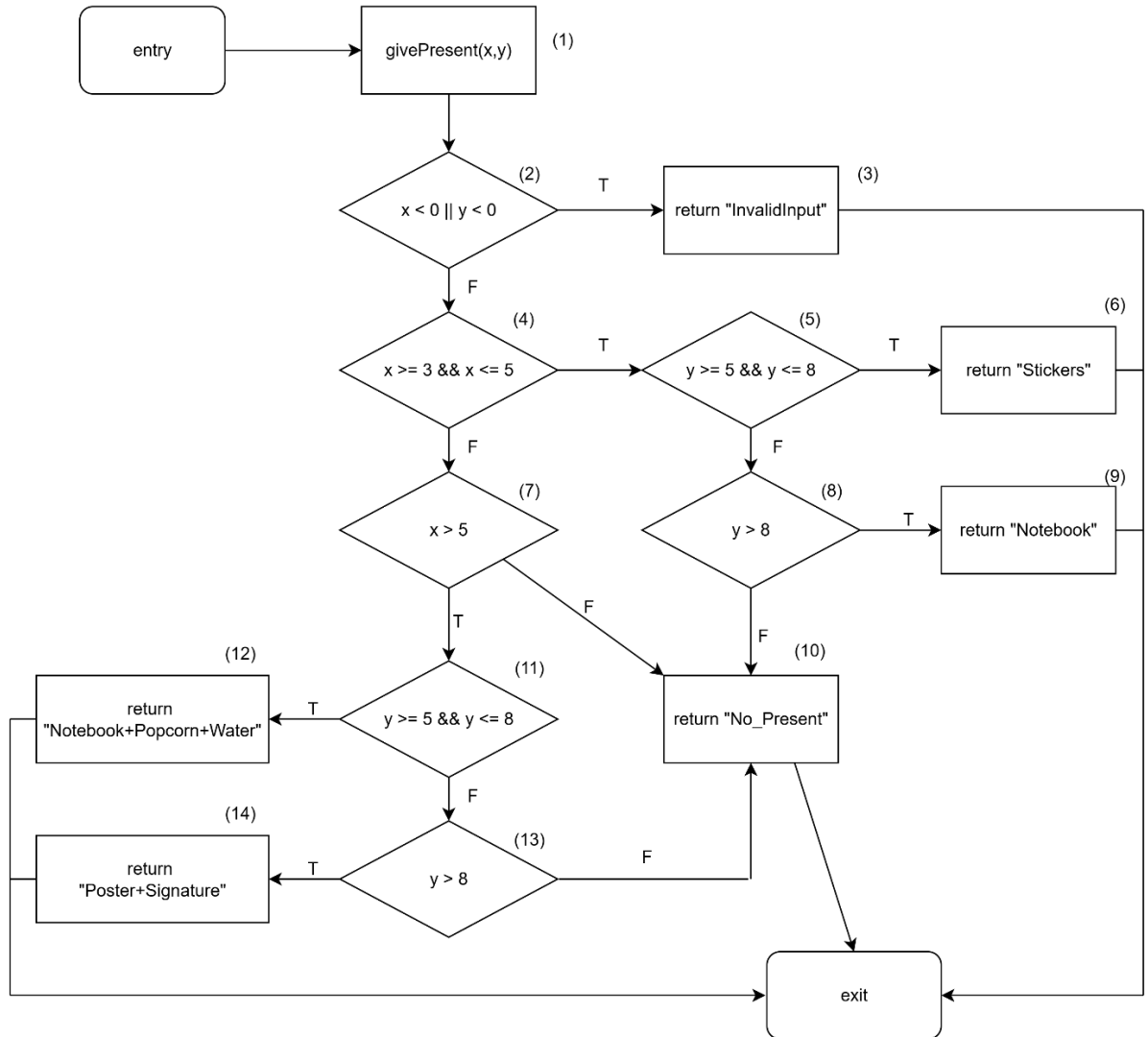
**Yêu cầu:** Báo cáo phân tích, thiết kế các ca kiểm thử, và kiểm thử chương trình của bạn với độ phủ All-use.

**Giải:**

**Dựa vào đồ thị dòng điều khiển, ta có:**

$$\text{def}(x) = \{1\}, \text{p-use}(x) = \{2, 4, 7\}, \text{c-use}(x) = \{\}$$

$$\text{def}(y) = \{1\}, \text{p-use}(y) = \{2, 5, 8, 11, 13\}, \text{c-use}(y) = \{\}$$



Variable	du-pair	def-clear-path	complete path	testcase
x	(1, 2T)	1, 2T	1, 2T, 3	(-1, 5)
	(1, 2F)	1, 2F	1, 2F, 4F, 7F, 10	(2, 1)
	(1, 4F)	1, 2F, 4F	1, 2F, 4F, 7F, 10	(2, 1)
	(1, 4T)	1, 2F, 4T	1, 2F, 4T, 5T, 6	(3, 5)
	(1, 7F)	1, 2F, 4F, 7F	1, 2F, 4F, 7F, 10	(2, 1)
	(1, 7F)	1, 2F, 4F, 7T	1, 2F, 4F, 7T, 11T, 12	(6, 6)
	(1, 7F)	1, 2F, 4F, 7T	1, 2F, 4F, 7T, 11T, 12	(6, 6)
y	(1, 2T)	1, 2T	1, 2T, 3	(-1, 5)
	(1, 2F)	1, 2F	1, 2F, 4F, 7F, 10	(2, 1)
	(1, 5T)	1, 2F, 4T, 5T	1, 2F, 4T, 5T, 6	(3, 5)
	(1, 5F)	1, 2F, 4T, 5F	1, 2F, 4T, 5F, 8T, 9	(5, 9)
	(1, 8T)	1, 2F, 4T, 5F, 8T	1, 2F, 4T, 5F, 8T, 9	(5, 9)
	(1, 8F)	1, 2F, 4T, 5F, 8F	1, 2F, 4T, 5F, 8F, 10	(5, 4)
	(1, 11T)	1, 2F, 4F, 7T, 11T	1, 2F, 4F, 7T, 11T, 12	(6, 6)
	(1, 11F)	1, 2F, 4F, 7T, 11F	1, 2F, 4F, 7T, 11F, 13T, 14	(10, 10)
	(1, 13T)	1, 2F, 4F, 7T, 11F, 13T	1, 2F, 4F, 7T, 11F, 13T, 14	(10, 10)
	(1, 13F)	1, 2F, 4F, 7T, 11F, 13F	1, 2F, 4F, 7T, 11F, 13F, 10	(6, 0)
	(1, 13F)	1, 2F, 4F, 7T, 11F, 13F	1, 2F, 4F, 7T, 11F, 13F, 10	(6, 0)
	(1, 13F)	1, 2F, 4F, 7T, 11F, 13F	1, 2F, 4F, 7T, 11F, 13F, 10	(6, 0)
	(1, 13F)	1, 2F, 4F, 7T, 11F, 13F	1, 2F, 4F, 7T, 11F, 13F, 10	(6, 0)

ID	Input		Expected	Actual	Result
	x	y			
L1	-1	5	Input invalid	Input invalid	TRUE
L2	2	1	Không quà	Không quà	TRUE
L3	3	5	Stickers	Stickers	TRUE
L4	6	6	Vở + Ngô + Nước	Vở + Ngô + Nước	TRUE
L5	1	-5	Input invalid	Input invalid	TRUE
L6	5	9	Vở	Vở	TRUE
L7	5	4	Không quà	Không quà	TRUE
L8	10	10	Poster + Chữ ký	Poster + Chữ ký	TRUE
L9	6	0	Không quà	Không quà	TRUE

(9/9)

[Link Github](#)