

ĐẠI HỌC CÔNG NGHỆ - ĐẠI HỌC QUỐC GIA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP

**ĐỀ BÀI: TÍNH GIÁ VÉ XEM PHIM CỦA MỖI NGƯỜI THEO ƯU
ĐÃI ĐƯỢC NHẬN**

Tên môn học: Kiểm thử và đảm bảo chất lượng phần mềm

Mã lớp học: INT3117_3

Giảng viên: ThS. Nguyễn Thu Trang

Sinh viên thực hiện: Nguyễn Đăng Đạo - 23021516

GitHub: [software_testing_and_quality_assurance](https://github.com/SoftwareTestingAndQualityAssurance/software_testing_and_quality_assurance)

HÀ NỘI - 2025

Mục lục

1. Giới thiệu	3
2. Phân tích chương trình	3
3. Đồ thị dòng điều khiển	4
4. Kiểm thử dòng dữ liệu	5
5. Bài tập trong slide	8

1. Giới thiệu

Hàm `tinhGiaVe(int tuoi, String ngay, boolean laHocSinh, boolean laThanhVien)` được sử dụng để tính giá vé xem phim dựa trên các yếu tố:

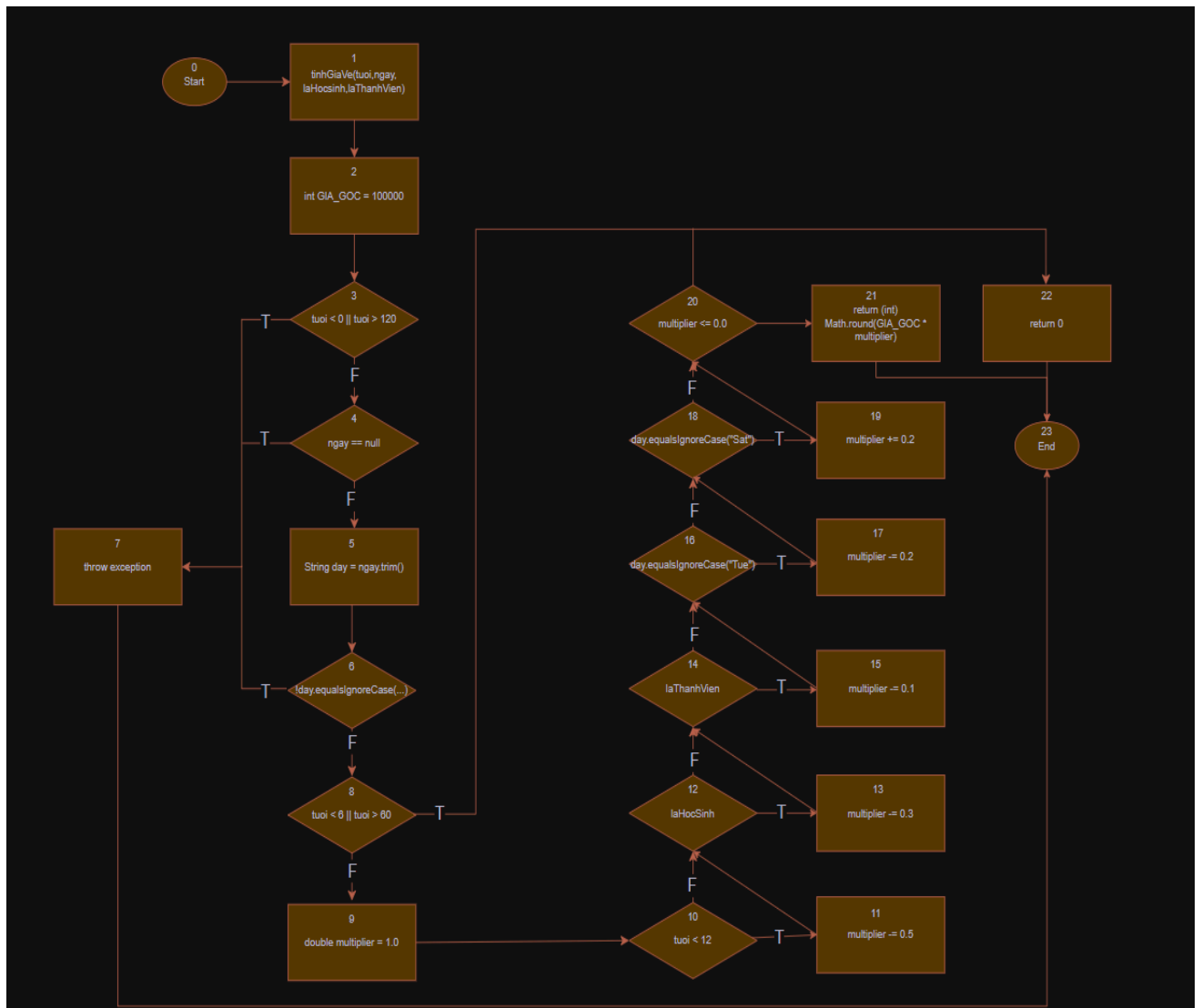
- Tuổi của khách hàng
- Ngày xem phim
- Tình trạng học sinh
- Tình trạng thành viên rạp

Mục tiêu của bài này là phân tích luồng điều khiển, xác định các nhánh logic, và thiết kế bộ ca kiểm thử đảm bảo độ bao phủ nhánh (C2) hay là mỗi điều kiện trong câu lệnh rẽ nhánh `if` đều được thực hiện ít nhất một lần với giá trị `True` và `False`

2. Phân tích chương trình

STT	Câu điều kiện	Ý nghĩa	Hành vi khi TRUE	Hành vi khi FALSE
1	<code>tuoi < 0 tuoi > 120</code>	Tuổi không hợp lệ	Ném ngoại lệ	Tiếp tục
2	<code>(ngay == null)</code>	Kiểm tra ngày hợp lệ	Ném ngoại lệ	Tiếp tục
3	<code>!(Mon/Tue/.../Sun)</code>	Ngày không hợp lệ	Ném ngoại lệ	Tiếp tục
4	<code>tuoi < 6 tuoi >= 60</code>	Miễn phí vé cho trẻ nhỏ và người già	Trả về 0	Tiếp tục
5	<code>tuoi < 12</code>	Giảm 50% cho trẻ nhỏ	Giảm multiplier 0.5	Tiếp tục
6	<code>(laHocSinh)</code>	Giảm 30% cho học sinh	Giảm multiplier 0.3	Tiếp tục
7	<code>(laThanhVien)</code>	Giảm 10% cho thành viên	Giảm multiplier 0.1	Tiếp tục
8	<code>(day.equalsIgnoreCase("Tue"))</code>	Giảm thêm 20% vào thứ Ba	Giảm multiplier 0.2	Tiếp tục
9	<code>(day.equalsIgnoreCase("Sat"))</code>	Tăng 20% vào thứ Bảy	Tăng multiplier 0.2	Tiếp tục
10	<code>(multiplier <= 0.0)</code>	Nếu giảm quá nhiều, vé miễn phí	Trả về 0	Tính giá vé bình thường

3. Đồ thị dòng điều khiển



4. Kiểm thử dòng dữ liệu

a. Các biến:

Biến	def	p-use	c-use
tuoi	1	3, 8, 10	-
ngay	1	4	5
day	5	6, 16, 18	-
laHocSinh	1	12	-
laThanhVien	1	14	-
multiplier	9, 11, 13, 15, 17, 19	20	21
GIA_GOC	2	-	21

b. Data flow

Biến	DU-pair	Def-clear path	All-defs	All-p-uses	All-c-uses	All-p-uses/some-c-use	All-c-uses/some-p-use	All-uses	All-du-pairs
tuoi	(1,3)	1,2,3	X	X		X		X	X
tuoi	(1,8)	1,2,3F,4F,5,6F,8		X		X		X	X
tuoi	(1,10)	1,2,3F,4F,5,6F,8F,9,10		X		X		X	X
ngay	(1,4)	1,2,3F,4	X	X			X	X	X
ngay	(1,5)	1,2,3F,4F,5			X		X	X	X
day	(5,6)	5,6	X	X		X		X	X
day	(5,16)	5,6F,8F,9,10F,12F,14F,16		X		X		X	X
		5,6F,8F,9,10F,12F,14T,15,16		X		X		X	X
		5,6F,8F,9,10F,12T,13,14F,16		X		X		X	X
		5,6F,8F,9,10F,12T,13,14T,15,16		X		X		X	X
		5,6F,8F,9,10T,12F,14F,16		X		X		X	X
		5,6F,8F,9,10T,12F,14T,15,16		X		X		X	X
		5,6F,8F,9,10T,12T,13,14F,16		X		X		X	X
		5,6F,8F,9,10T,12T,13,14T,15,16		X		X		X	X
day	(5,18)	5,6F,8F,9,10F,12F,14F,16T,18		X		X		X	X
		5,6F,8F,9,10F,12F,14T,15,16T,18		X		X		X	X
		5,6F,8F,9,10F,12T,13,14F,16T,18		X		X		X	X
		5,6F,8F,9,10F,12T,13,14T,15,16T,18		X		X		X	X
		5,6F,8F,9,10T,12F,14F,16T,18		X		X		X	X
		5,6F,8F,9,10T,12F,14T,15,16T,18		X		X		X	X
		5,6F,8F,9,10T,12T,13,14F,16T,18		X		X		X	X
		5,6F,8F,9,10T,12T,13,14T,15,16T,18		X		X		X	X
laHocSinh	(1,12)	1,2,3F,4F,5,6F,8F,9,10F,12	X	X		X		X	X
		1,2,3F,4F,5,6F,8F,9,10T,11,12	X	X		X		X	X
laThanhVien	(1,14)	1,2,3F,4F,5,6F,8F,9,10F,12F,14	X	X		X		X	X
		1,2,3F,4F,5,6F,8F,9,10F,12T,13,14	X	X		X		X	X
		1,2,3F,4F,5,6F,8F,9,10T,11,12T,13,14	X	X		X		X	X
		1,2,3F,4F,5,6F,8F,9,10T,11,12F,14	X	X		X		X	X
multiplier	(9,20)	9,10F,12F,14F,16F,18F,20	X	X		X		X	X
multiplier	(11,20)	11,12F,14F,16F,18F,20	X	X		X		X	X
multiplier	(13,20)	13,14F,16F,18F,20	X	X		X		X	X
multiplier	(15,20)	15,16F,18F,20	X	X		X		X	X
multiplier	(17,20)	17,18F,20	X	X		X		X	X
multiplier	(19,20)	19,20	X	X		X		X	X

c. Bảng test case tương ứng

ID	tuoi	ngay	hocsinh	thanhvien	Giá vé mong đợi
1	-1	Mon	FALSE	FALSE	Exception
2	6	Tue	FALSE	FALSE	30000
3	11	Mon	FALSE	FALSE	50000
4	30	Teu	FALSE	FALSE	Exception
5	30	Teu	FALSE	FALSE	Exception
6	30	Teu	FALSE	FALSE	Exception
7	30	Tue	FALSE	FALSE	80000
8	30	Tue	FALSE	TRUE	70000
9	30	Tue	TRUE	FALSE	50000
10	30	Tue	TRUE	TRUE	40000
11	10	Tue	FALSE	FALSE	30000
12	10	Tue	FALSE	TRUE	20000
13	10	Tue	TRUE	FALSE	0
14	10	Tue	TRUE	TRUE	0
15	30	Sat	FALSE	FALSE	120000
16	30	Sat	FALSE	TRUE	110000
17	30	Sat	TRUE	FALSE	90000
18	30	Sat	TRUE	TRUE	80000
19	10	Sat	FALSE	FALSE	70000
20	10	Sat	FALSE	TRUE	60000
21	10	Sat	TRUE	FALSE	40000
22	10	Sat	TRUE	TRUE	30000
23	30	Mon	FALSE	FALSE	100000
24	30	Mon	FALSE	FALSE	100000
25	10	Mon	FALSE	FALSE	50000
26	10	Mon	TRUE	FALSE	20000
27	30	Mon	TRUE	FALSE	70000
28	30	Mon	FALSE	FALSE	100000
29	30	Mon	FALSE	FALSE	100000
30	11	Mon	FALSE	FALSE	50000
31	30	Mon	TRUE	FALSE	70000
32	30	Mon	FALSE	TRUE	90000
33	30	Sat	FALSE	FALSE	120000
34	30	Mon	FALSE	FALSE	100000

d. Kết quả kiểm thử

```
Test case 1: lỗi từ hàm tínhGiaVe: Tuổi không hợp lệ: -1
Test case 2: tuoi=6, ngay=Tue, hocSinh=false, thanhVien=false => giaVe=30000, expected=30000 => PASS
Test case 3: tuoi=11, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=50000, expected=50000 => PASS
Test case 4: lỗi từ hàm tínhGiaVe: Ngày không hợp lệ: Teu
Test case 5: lỗi từ hàm tínhGiaVe: Ngày không hợp lệ: Teu
Test case 6: lỗi từ hàm tínhGiaVe: Ngày không hợp lệ: Teu
Test case 7: tuoi=30, ngay=Tue, hocSinh=false, thanhVien=false => giaVe=80000, expected=80000 => PASS
Test case 8: tuoi=30, ngay=Tue, hocSinh=false, thanhVien=true => giaVe=70000, expected=70000 => PASS
Test case 9: tuoi=30, ngay=Tue, hocSinh=true, thanhVien=false => giaVe=50000, expected=50000 => PASS
Test case 10: tuoi=30, ngay=Tue, hocSinh=true, thanhVien=true => giaVe=40000, expected=40000 => PASS
Test case 11: tuoi=10, ngay=Tue, hocSinh=false, thanhVien=false => giaVe=30000, expected=30000 => PASS
Test case 12: tuoi=10, ngay=Tue, hocSinh=false, thanhVien=true => giaVe=20000, expected=20000 => PASS
Test case 13: tuoi=10, ngay=Tue, hocSinh=true, thanhVien=false => giaVe=0, expected=0 => PASS
Test case 14: tuoi=10, ngay=Tue, hocSinh=true, thanhVien=true => giaVe=0, expected=0 => PASS
Test case 15: tuoi=30, ngay=Sat, hocSinh=false, thanhVien=false => giaVe=120000, expected=120000 => PASS
Test case 16: tuoi=30, ngay=Sat, hocSinh=false, thanhVien=true => giaVe=110000, expected=110000 => PASS
Test case 17: tuoi=30, ngay=Sat, hocSinh=true, thanhVien=false => giaVe=90000, expected=90000 => PASS
Test case 18: tuoi=30, ngay=Sat, hocSinh=true, thanhVien=true => giaVe=80000, expected=80000 => PASS
Test case 19: tuoi=10, ngay=Sat, hocSinh=false, thanhVien=false => giaVe=70000, expected=70000 => PASS
Test case 20: tuoi=10, ngay=Sat, hocSinh=false, thanhVien=true => giaVe=60000, expected=60000 => PASS
Test case 21: tuoi=10, ngay=Sat, hocSinh=true, thanhVien=false => giaVe=40000, expected=40000 => PASS
Test case 22: tuoi=10, ngay=Sat, hocSinh=true, thanhVien=true => giaVe=30000, expected=30000 => PASS
Test case 23: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=100000, expected=100000 => PASS
Test case 24: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=100000, expected=100000 => PASS
Test case 25: tuoi=10, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=50000, expected=50000 => PASS
Test case 26: tuoi=10, ngay=Mon, hocSinh=true, thanhVien=false => giaVe=20000, expected=20000 => PASS
Test case 27: tuoi=30, ngay=Mon, hocSinh=true, thanhVien=false => giaVe=70000, expected=70000 => PASS
Test case 28: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=100000, expected=100000 => PASS
Test case 29: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=100000, expected=100000 => PASS
Test case 30: tuoi=11, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=50000, expected=50000 => PASS
Test case 31: tuoi=30, ngay=Mon, hocSinh=true, thanhVien=false => giaVe=70000, expected=70000 => PASS
Test case 32: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=true => giaVe=90000, expected=90000 => PASS
Test case 33: tuoi=30, ngay=Sat, hocSinh=false, thanhVien=false => giaVe=120000, expected=120000 => PASS
Test case 34: tuoi=30, ngay=Mon, hocSinh=false, thanhVien=false => giaVe=100000, expected=100000 => PASS
```

Kết luận: Chương trình pass toàn bộ test case. Chưa phát hiện ra lỗi

5. Bài tập trong slide

BT1: Trình bày các bước trong quy trình kiểm thử dòng dữ liệu động

Bước	Nội dung	Kết quả mong đợi
1	Phân tích mã nguồn & biến	Danh sách biến, điểm def/use
2	Vẽ CFG	Biểu đồ luồng điều khiển
3	Xác định def-use & def-clear path	Danh sách DU-pair
4	Chọn tiêu chí kiểm thử	Mức bao phủ mong muốn
5	Xác định đường đi	Danh sách đường đi cần test
6	Sinh test case	Bộ dữ liệu đầu vào
7	Thực thi kiểm thử	Kết quả thực tế vs mong đợi
8	Đánh giá độ phủ	Hoàn thiện kiểm thử

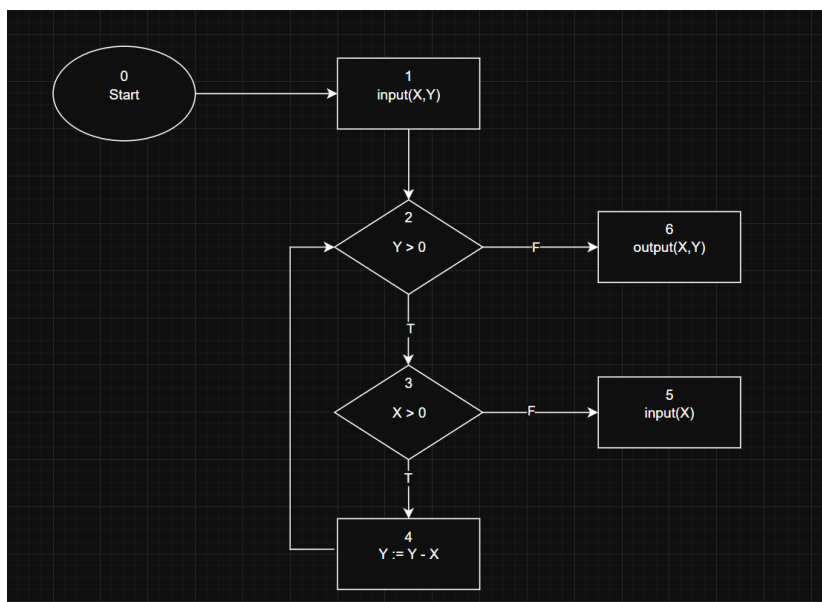
BT2:

Bài 2:

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



Biến	def	p-use	c-use
X	1, 5	3	4, 7
Y	1, 4	2	4, 7

Biến	DU-pair	Def-clear path	Test case
X	(1,3)	1 - 2(T) - 3	X=2, Y=2
	(1,4)	1 - 2(T) - 3(T) - 4	X=2, Y=2
	(1,7)	1 - 2(F) - 7	X=0, Y=0
	(5,3)	1 - 2(T) - 3(F) - 5 - 2(T) - 3	X=-1,Y=5
	(5,4)	1 - 2(T) - 3(F) - 5 - 2(T) - 3(T) - 4	X=-1,Y=5
	(5,7)	1 - 2(T) - 3(F) - 5 - 2(F) - 7	X=-1,Y=1
Y	(1,2)	1 - 2	X=0, Y=0
	(1,4)	1 - 2(T) - 3(T) - 4	X=2, Y=2
	(1,7)	1 - 2(F) - 7	X=0, Y=0
	(4,2)	1 - 2(T) - 3(T) - 4 - 2	X=-1,Y=5
	(4,4)	1 - 2(T) - 3(T) - 4 - 2(T) - 3(T) - 4	X=1,Y=5
	(4,7)	1 - 2(T) - 3(T) - 4 - 2(F) - 7	X=2, Y=2

BT3:

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

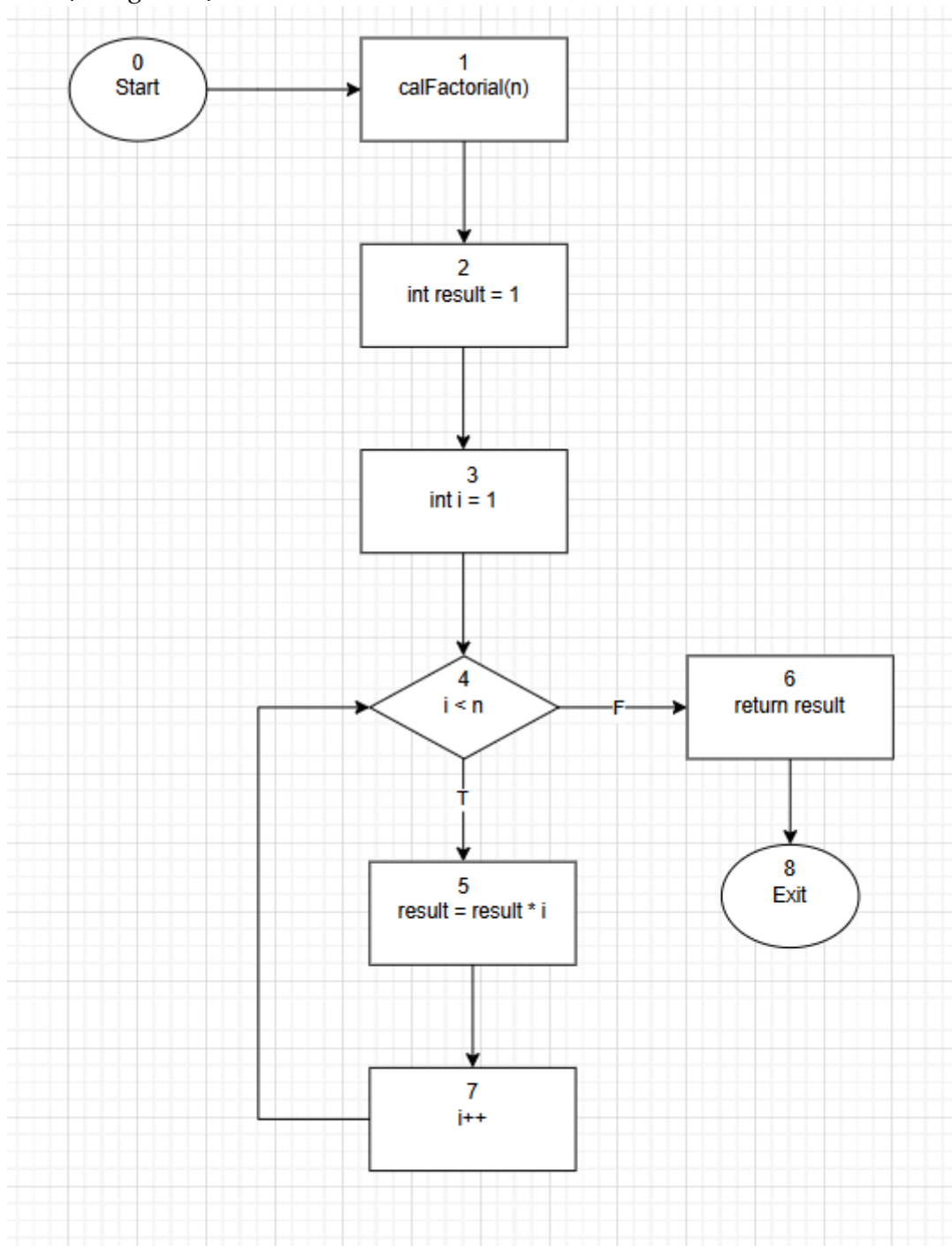
- Hãy liệt kê các câu lệnh ứng với các khái niệm *def*, *c-use*, và *p-use* ứng với các biến được sử dụng trong hàm này.
- Hãy vẽ đồ thị dòng dữ liệu của hàm này.

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

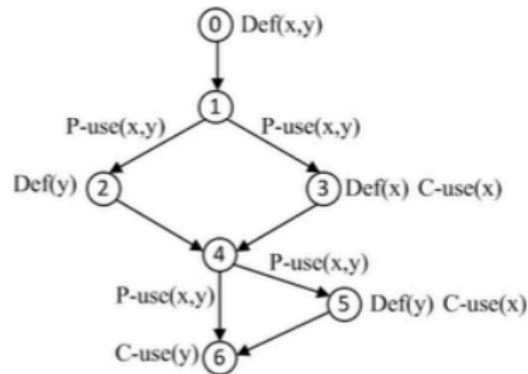
```
int calFactorial (int n){
    int result = 1;
    int i=1;
    while (i <= n){
        result = result *i;
        i++;
    }//end while
    return result;
}//the end
```

Biến	def	p-use	c-use
n	calFactorial(n)	$i \leq n$	none
result	int result = 1, result = result + 1	none	result = result + 1, return result
i	int i = 1, i++	$i \leq n$	i++

Đồ thị dòng dữ liệu



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*?

Biến	def	p-use	c-use
x	0,3	1,4,(1,3),(4,5)	3,5
y	0,2,5	1,4,(1,3),(4,5)	6

1,

Biến	du pair	def- clear path
x	(0,1)	0 - 1
	(0,3)	0 - 1 - 3
	(0,4)	0 - 1 - 2 - 4
	(0,5)	0 - 1 - 2 - 4 - 5
	(3,4)	3 - 4
	(3,5)	3 - 5

y	(0,1)	0 - 1
	(0,4)	0 - 1 - 3 - 4
	(0,6)	0 - 1 - 3 - 4 -6
	(2,4)	2 - 4
	(2,6)	2 - 4 -6
	(5,6)	5 - 6

2,

Biến	du pair
x	(0,1), (0,3), (0,4), (0,5),(3,4), (3,5)
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	(0,1)	0 - 1	0 - 1 - 3 - 4 - 6
	(0, 1-3)	0 - 1 - 3	0 - 1 - 3 - 4 - 5 - 6
	(0,4)	0 - 1 - 2 - 4	0 - 1 - 2 - 4 - 6
	(0, 4-5)	0 - 1 - 2 - 4 - 5	0 - 1 - 2 - 4 - 5 - 6
	(3,4)	3 - 4	0 - 1 - 3 - 4 - 6
	(3, 4-5)	3 - 4 - 5	0 - 1 - 3 - 4 - 5 - 6
y	(0,1)	0 - 1	0 - 1 - 3 - 4 - 6
	(0,4)	0 - 1 - 3 - 4	0 - 1 - 3 - 4 - 6
	(0, 1-3)	0 - 1 - 3	0 - 1 - 3 - 4 - 5 - 6
	(0, 4-5)	0 - 1 - 3 - 4 - 5	0 - 1 - 3 - 4 - 5 - 6
	(2,4)	2 - 4	0 - 1 - 2 - 4 - 6
	(2, 4-5)	2 - 4 - 5	0 - 1 - 2 - 4 - 5 - 6

All-c-use/ some-p-use

Variable	Du-pair	Def-clear path	Complete path
x	(0,3)	0 - 1 - 3	0 - 1 - 3 - 4 - 6
	(0,5)	0 - 1 - 2 - 4 - 5	0 - 1 - 2 - 4 - 5 - 6
	(3,5)	3 - 5	0 - 1 - 3 - 4 - 5 - 6
y	(0,6)	0 - 1 - 3 - 4 - 6	0 - 1 - 3 - 4 - 6
	(2,6)	2 - 4 - 6	0 - 1 - 2 - 4 - 6
	(5,6)	5 - 6	0 - 1 - 3 - 4 - 5 - 6

4, (1,3): $x + y = 4 \Rightarrow y = 4 - x$ (1)

(4,5): $x^2 + y^2 > 17$ (2)

từ (1) và (2) $\Rightarrow x > 4,12$ hoặc $x < -0,12$

\Rightarrow có x thỏa mãn \Rightarrow tồn tại cặp (x,y) thỏa mãn (2) \Rightarrow đường đi khả thi

Ví dụ $x = -1 \Rightarrow y = 5 \Rightarrow x^2 + y^2 = 26 > 17$

5, Ở nút 3 biến x có chứa cả lệnh def(x) và c-use(x) nhưng không có cặp du - pair(3,3) vì ta cần một def clear path từ 3 đến 3. Ta có thể thấy def sẽ được thực thi trước c-use hay ta có 1 lệnh tương đương là $x = x + 1$ đây không là 1 du - pair vì vậy (3,3) không thể là du-pair

BT5:

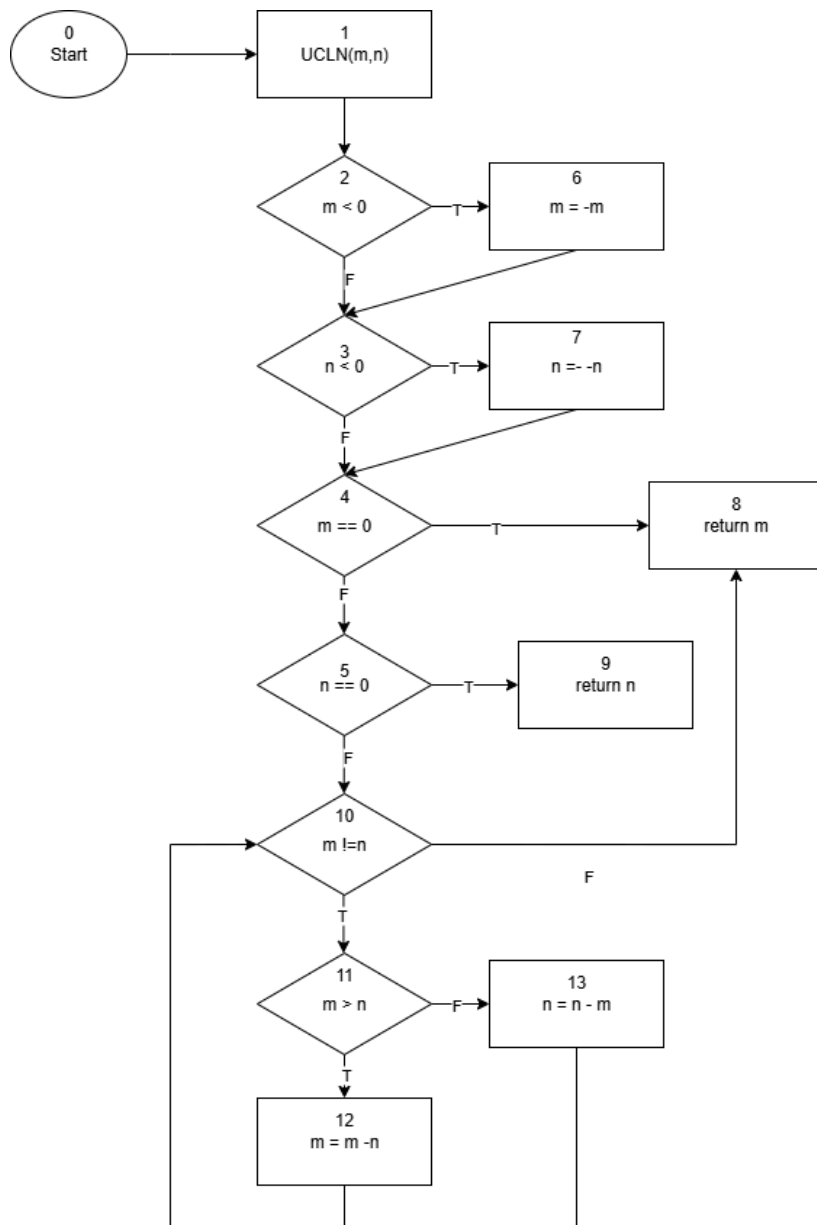
Bài 5

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;
}
```



Sinh đường phủ c2:

- Path 1: 0 - 1 - 2F - 3F - 4F - 5F - 10T - 11T - 12 - 10T - 11F - 13 - 10 F - 8
Test input: $m = 2, n = 1$
- Path 2: 0 - 1 - 2T - 6 - 3T - 7 - 4F - 5F - 10F - 8
Test input: $m = -1, n = -1$
- Path 3: Path 1: 0 - 1 - 2F - 3F - 4F - 5T - 9
Test input: $m = 1; n = 0$

All-Def:

Biến	def	p-use	c-use
m	1,6,12	2,4,10,11	6,8,12,13
n	1,7,13	3,5,10,11	7,9,12,13

Biến	du pair	complete path	test input
m	(1,2)	0 - 1 - 2F - 3F - 4F - 5T - 9	m = 1; n = 0
	(6,8)	0 - 1 - 2F - 3F - 4T - 8	m = 0, n = 1
	(12,13)	0 - 1 - 2F - 3F - 4F - 5F - 10T - 11T - 12 - 10T - 11F - 13 - 10 F - 8	m = 2, n = 1
n	(1,3)	0 - 1 - 2F - 3F - 4F - 5T - 9	m = 0, n = 1
	(7,10)	0 - 1 - 2T - 6 - 3T - 7 - 4F - 5F - 10F - 8	m = -1, n = -1
	(13,10)	0 - 1 - 2F - 3F - 4F - 5F - 10T - 11T - 12 - 10T - 11F - 13 - 10 F - 8	m = 2, n = 1