

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO BÀI TẬP

Sinh viên: Nguyễn Đức Duy
Mã sinh viên: 22028215
Lớp: INT3117_1

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

Bài làm:

- **Bước 1:** Xác định đơn vị chương trình cần kiểm thử (Chọn module hoặc hàm cụ thể)
- **Bước 2:** Xây dựng đồ thị dòng dữ liệu
- **Bước 3:** Xác định tiêu chí kiểm thử
- **Bước 4:** Chọn đường đi phù hợp với tiêu chí đã chọn
- **Bước 5:** Sinh các ca kiểm thử và thực hiện.

2.

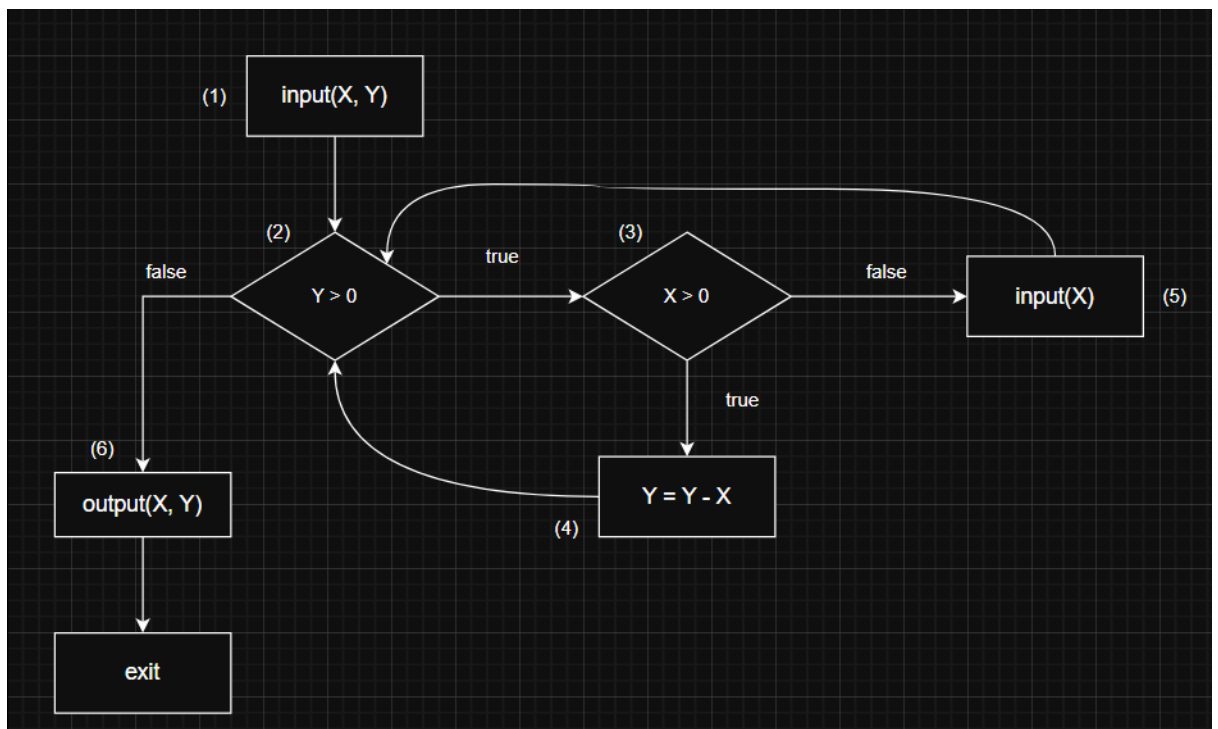
```

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



b.

Biến	Def	Du-pair
x	1	1,3(T) 1,3(F) 1,4 1,6

	5	5,3(T) 5,3(F) 5,4 5,6
Y	1	1,2(T) 1,2(F) 1,4 1,6
	4	4,2(T) 4,2(F) 4,4 4,6

c.

Biến x: def: 1, 5

p-use: 3

c-use: 4, 5

Biến y: def: 1, 4

p-use: 2

c-use: 4,6

Biến	Du-pair	Def-clear path	Complete path
X	1,3(T) 1,3(F) 1,4 1,6	1, 2(T), 3(T) 1, 2(T), 3(F) 1, 2(T), 3(T), 4 1, 2(F), 6	1, 2(T), 3(T), 4, 2(F), 6 1, 2(T), 3(F), 5, 2(F), 6 1, 2(T), 3(T), 4, 2(F), 6 1, 2(F), 6
	5,3(T) 5,3(F) 5,4 5,6	5, 2T, 3T 5, 2T, 3F 5, 2T, 3T, 4 5, 2F, 6	1, 2T, 3F, 5, 2T, 3T, 4, 2F, 6 1, 2T, 3F, 5, 2T, 3F, 5, 2F, 6 1, 2T, 3F, 5, 2T, 3T, 4, 2F, 6 1,2T, 3F, 5, 2F, 6
Y	1,2T 1,2F 1,4 1,6	1,2T 1,2F 1, 2T, 3T, 4 1, 2F, 6	1, 2T, 3T, 4, 2F, 6 1, 2F, 6 1, 2T, 3T, 4, 2F, 6 1, 2F, 6
	4,2T	4, 2T	1, 2T, 3T, 4, 2T, 3T, 4, 2F, 6

	4,2F 4,4 4,6	4, 2F 4, 2T, 3T, 4 4, 2F, 6	1, 2T, 3T, 4, 2F, 6 1, 2T, 3T, 4, 2T, 3T, 4, 2F, 6 1, 2T, 3T, 4, 2F, 6
--	--------------------	-----------------------------------	--

3.

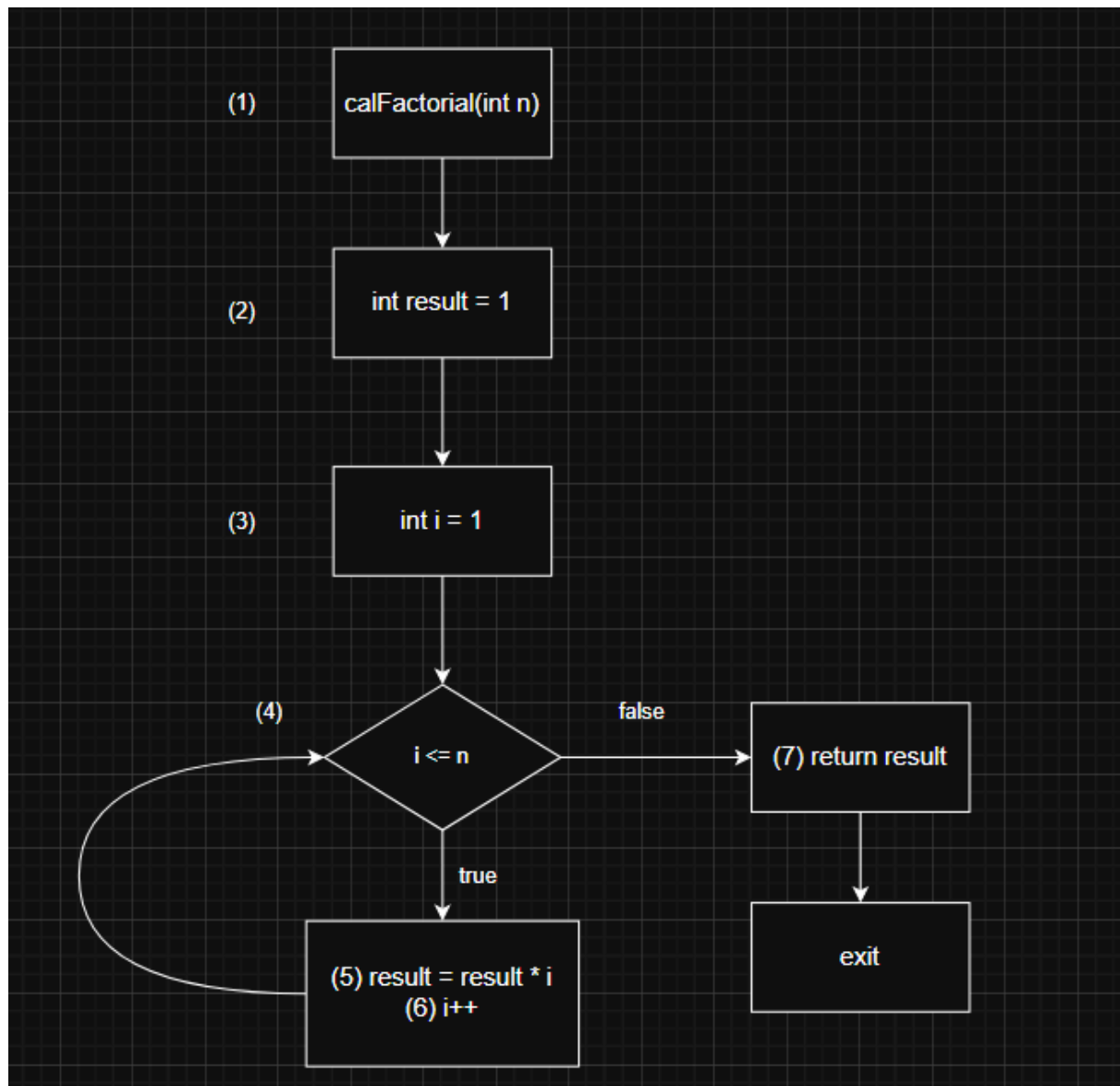
7. Cho hàm `calFactorial` viết bằng ngôn ngữ C như Đoạn mã 7.7.

- 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`

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

Bài làm



Biến n: def: 1

p-use: 4

c-use

Biến result: def: 2, 5

p-use:

c-use: 5, 7

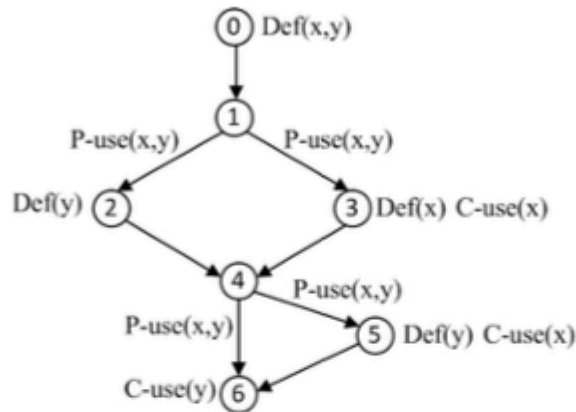
Biến i: def: 3, 6

p-use: 4

c-use: 5, 6

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*?

Biến x : def: 0, 3

p-use: 1, 4

c-use: 3, 5

Biến y : def: 0, 2, 5

p-use: 1, 4

c-use: 6

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

	c-use	def-clear path	complete path
x	0, 3 0, 5 3, 5	0, 1, 3 0, 1, 2, 4, 5 3, 4, 5	0, 1, 3, 4, 6 0, 1, 2, 4, 5, 6 0, 1, 3, 4, 5, 6
y	0, 6 2, 6 5, 6	0, 1, 3, 4, 6 2, 4, 6 5, 6	0, 1, 3, 4, 6 0, 1, 2, 4, 6 0, 1, 2, 4, 5, 6

	p-use	def-clear path	complete path
x	0, 1 0, 4 3, 4	0, 1, 2 0, 1, 3 0, 1, 2, 4, 5 0, 1, 2, 4, 6 3, 4, 5 3, 4, 6	0, 1, 2, 4, 6 0, 1, 3, 4, 6 0, 1, 2, 4, 5, 6 0, 1, 2, 4, 6 0, 1, 3, 4, 5, 6 0, 1, 3, 4, 6
y	0, 1	0, 1, 2 0, 1, 3	0, 1, 2, 4, 6 0, 1, 3, 4, 6

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

Path: 0 - 1 - 3 - 4 - 5 - 6 có thể thực thi được bởi vì:

Ta có hệ phương trình: $x + y = 4$ và $x^2 + y^2 > 17$ có vô số nghiệm

Định 3 có định nghĩa và sử dụng biến x nhưng không tồn tại mối quan hệ def - use bởi vì thứ tự thực thi: biến x được sử dụng trước, còn định nghĩa sau.

5.

Bài 5

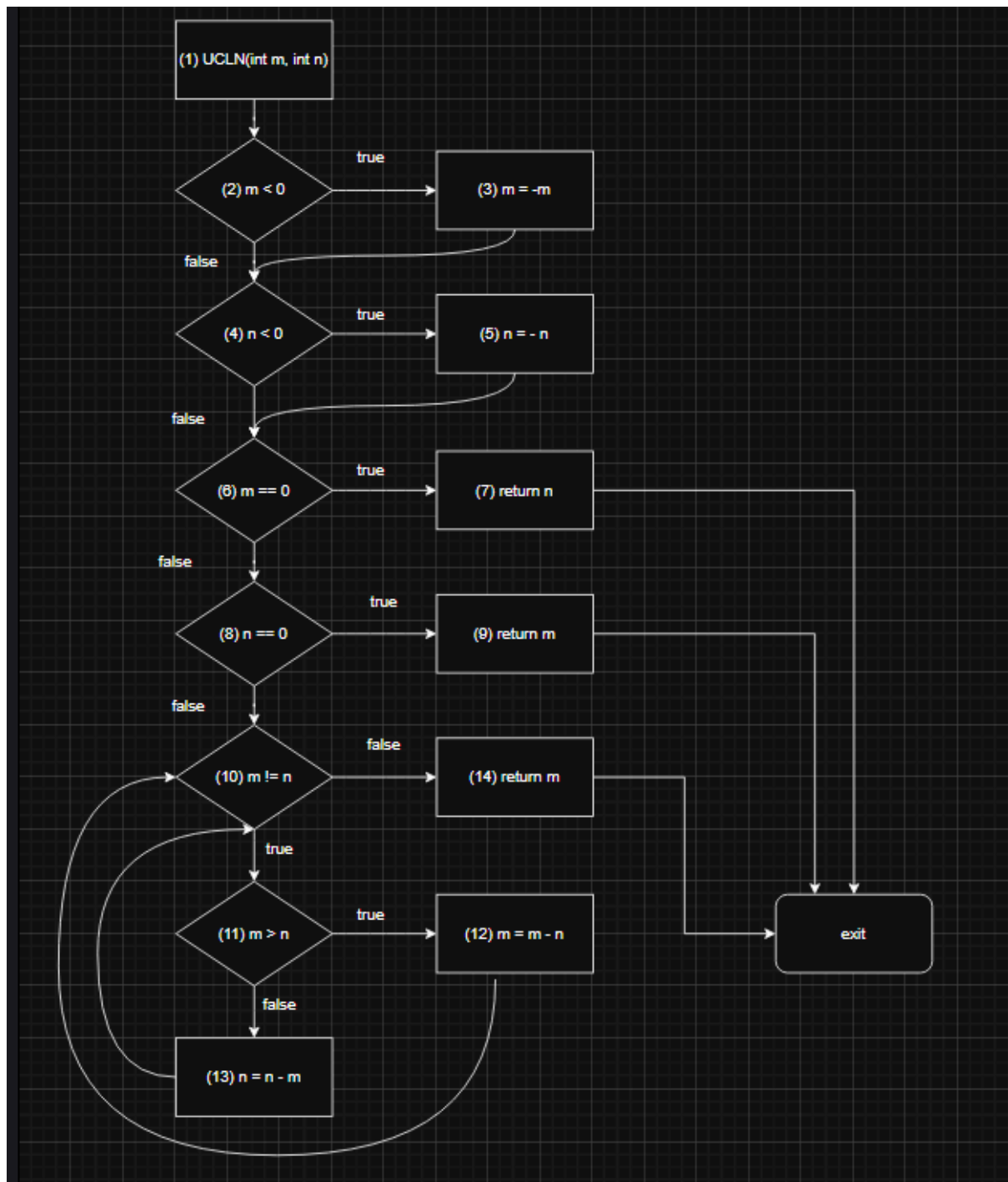
Cho đoạn mã nguồn như hình bên,

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.



2.

Path 1: 1, 2T, 3, 4T, 5, 6T, 7

=> Test case: UCLN(0, -1)

Path 2: 1, 2(F), 4(F), 6(F), 8(T), 9

=> Test case: UCLN(1, 0)

Path 3: 1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(F), 13, 10(F), 14

=> Test case: UCLN(6, 8)

Path 4: 1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(T), 12, 10(F), 14

=> Test case: UCLN(8, 6)

3. Biến m: def: 1, 3, 12

p-use: 2, 6, 10, 11

c-use: 3, 9, 14, 12, 13

Biến n: def: 1, 5, 13

p-use: 4, 8, 10, 11

c-use: 5, 7, 12, 13

	Du-pair	Def-clear path	complete path
m	1, 3 3, 9 12,14	1, 2T, 3 3, 4F, 6F, 8T, 9 12, 10F, 14	1, 2T, 3, 4F, 6T, 7 1, 2T, 3, 4F, 6F, 8T, 9 1, 2F, 4F, 6F, 8F, 10T, 11T, 12, 10F, 14
n	1, 5 5, 7 13, 12	1, 2F, 4T, 5 5, 6T, 7 13, 10T, 11T, 12	1, 2F, 4T, 5, 6T, 7 1, 2F, 4T, 5, 6T, 7 1, 2F, 4F, 6F, 8F, 10T, 11F, 13, 10T, 11T, 12, 10F, 14

6. [Link mã nguồn](#)

6.1. Mô tả bài toán.

Hệ thống tính giá vé xem phim cho 1 người dựa vào Tuổi (Age) và Loại ngày (DayType)

- Nếu Age không hợp lệ (âm hoặc >100) -> báo lỗi
- Ngày chia 2 loại: ngày thường hoặc cuối tuần

- Nhóm tuổi:
 - Child: 0-12
 - Adult: 13-60
 - Senior: 60-100

Bảng giá:

- Child: Ngày thường = 50,000 VND, Cuối tuần = 60,000 VND
- Adult: Ngày thường = 100,000 VND, Cuối tuần = 120,000 VND
- Senior: Ngày thường = 70,000 VND, Cuối tuần = 80,000 VND

Đầu vào: Một số nguyên x, với $x \in [0..100]$ và một string là

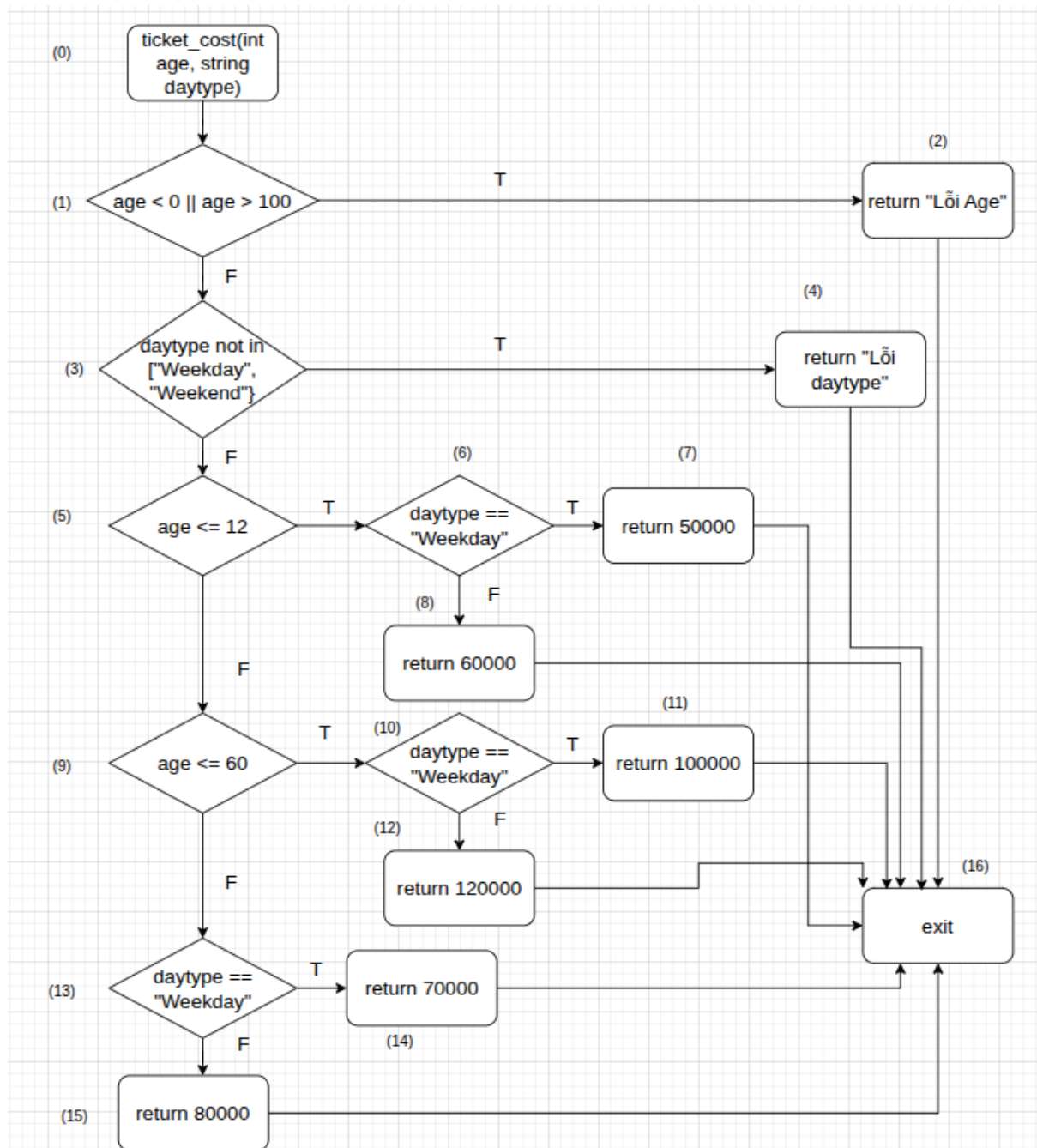
“Weekday” hoặc “Weekend”

Đầu ra: Giá vé hoặc báo lỗi.

6.2. Cài đặt chương trình

```
tem chi chuc nang > ticket.py > ...
1  def ticket_cost(age: int, daytype: str):
2      if age < 0 or age > 100:
3          return "Lỗi Age"
4      if daytype not in ["Weekday", "Weekend"]:
5          return "Lỗi DayType"
6
7      if age <= 12: #Child
8          return 50000 if daytype == "Weekday" else 60000
9      elif age <= 60: #Adult
10         return 100000 if daytype == "Weekday" else 120000
11     else: #Senior
12         return 70000 if daytype == "Weekday" else 80000
13
```

6.3 Xây dựng đồ thị dòng dữ liệu cho hàm ticket_cost



6.4 Kiểm thử với độ phủ all-uses

Biến age: def: 0

p-use: 1, 5, 9

c-use:

Biến daytype: def: 0

p-use: 3, 6, 10, 13

c-use:

Biến	Du-pair	Def-clear path	complete path
age	0, 1T 0, 1F 0, 5T 0, 5F 0, 9T 0, 9F	0, 1T 0, 1F 0, 1F, 3F, 5T 0, 1F, 3F, 5F 0, 1F, 3F, 5F, 9T 0, 1F, 3F, 5F, 9F	0, 1T, 2 0, 1F, 3T, 4 0, 1F, 3F, 5T, 6T, 7 0, 1F, 3F, 5F, 9T, 10T, 11 0, 1F, 3F, 5F, 9T, 10T, 11 0, 1F, 3F, 5F, 9F, 13T, 14
day type	0, 3T 0,3F 0,6T 0,6F 0, 10T 0,10F 0,13T 0,13F	0, 1F, 3T 0, 1F, 3F 0, 1F, 3F, 5T, 6T 0, 1F, 3F, 5T, 6F 0, 1F, 3F, 5F, 9T, 10T 0, 1F, 3F, 5F, 9T, 10F 0, 1F, 3F, 5F, 9F, 13T 0, 1F, 3F, 5F, 9F, 13F	0, 1F, 3T, 4 0, 1F, 3F, 5T, 6T, 7 0, 1F, 3F, 5T, 6T, 7 0, 1F, 3F, 5T, 6F, 8 0, 1F, 3F, 5F, 9T, 10T,11 0, 1F, 3F, 5F, 9T, 10T, 12 0, 1F, 3F, 5F, 9F, 13T, 14 0, 1F, 3F, 5F, 9F, 13F, 15

Các test case:

- 0, 1T, 2
- 0, 1F, 3T, 4
- 0, 1F, 3F, 5T, 6T, 7
- 0, 1F, 3F, 5F, 9T, 10T, 11
- 0, 1F, 3F, 5F, 9F, 13T, 14
- 0, 1F, 3F, 5T, 6F, 8
- 0, 1F, 3F, 5F, 9T, 10T, 12
- 0, 1F, 3F, 5F, 9F, 13F, 15

6.5 Kết quả kiểm thử bằng pytest

TC ID	Age	Daytype	EO	Result
1	-5	Weekday	Invalid Age	PASS
2	5	hello	Invalid Daytype	PASS
3	5	Weekday	50000	PASS
4	30	Weekday	100000	PASS

5	80	Weekday	70000	PASS
6	5	Weekend	60000	PASS
7	30	Weekend	120000	PASS
8	80	Weekend	80000	PASS