

BÁO CÁO BÀI TẬP: KIỂM THỬ DÒNG ĐIỀU KHIỂN

Môn học: Kiểm thử và đảm bảo chất lượng phần mềm (INT3117)

Họ và tên sinh viên: Nguyễn Chí Trung

Mã sinh viên: 22028075

1 Bài 1

Trình bày các bước nhằm kiểm thử một đơn vị chương trình theo phương pháp kiểm thử dòng điều khiển với một độ đo kiểm thử cho trước



Hình 1: Quy trình kiểm thử đơn vị chương trình dựa trên độ đo. **Nguồn:** Giáo trình Kiểm thử

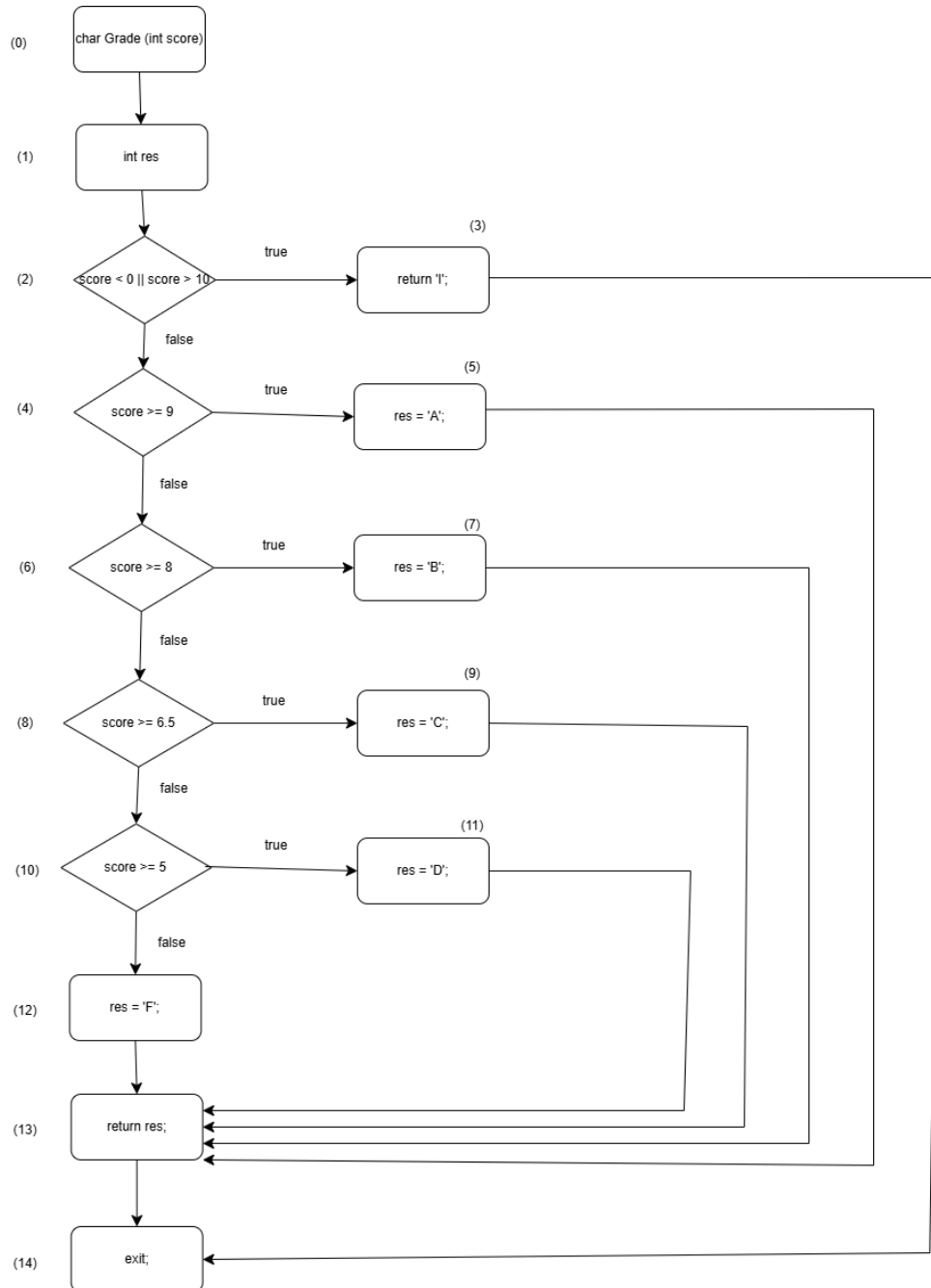
Để thực hiện việc kiểm thử một đơn vị chương trình theo phương pháp kiểm thử dòng điều khiển với một độ đo kiểm thử cho trước, ta cần thực hiện các bước sau:

- **Bước 1: Phân tích mã nguồn, xác định cấu trúc dòng điều khiển của chương trình ta cần kiểm thử:** Trong bước này, ta sẽ xác định các câu lệnh, các nhánh rẽ có thể xảy ra và vòng lặp của chương trình.
- **Bước 2: Xác định độ đo kiểm thử tương ứng từ yêu cầu của đề bài.** Ta xác định độ đo kiểm thử dựa trên các tiêu chí sau:
 - Phủ cấp 1 (C_1) - phủ câu lệnh: Tất cả các câu lệnh trong chương trình đều được thực thi tối thiểu 1 lần sau khi chạy tất cả các ca kiểm thử.
 - Phủ cấp 2 (C_2) - phủ nhánh: Đảm bảo rằng mỗi nhánh của chương trình sinh ra từ các cấu trúc điều kiện được thực thi tối thiểu một lần.
 - Phủ cấp 3 (C_3) - phủ điều kiện: Mỗi điều kiện con trong các biểu thức điều kiện phức tạp phải được kiểm thử cho trường hợp True và False.
 - Bao phủ vòng lặp: Xác định các vòng lặp có trong chương trình, rồi lần lượt kiểm thử với các trường hợp như lặp 0, 1, 2, k lần tương ứng.
- **Bước 3: Xây dựng đồ thị dòng điều khiển:** Đồ thị này cần biểu diễn được tất cả các đường đi có thể có trong chương trình, với các đỉnh thể hiện lệnh / khối lệnh, các cạnh biểu diễn luồng điều khiển của chương trình.
- **Bước 4: Xác định các đường đi kiểm thử độc lập của chương trình:** Xuất phát từ điểm bắt đầu (node entry), đi qua các đỉnh của đồ thị và kết thúc ở điểm kết thúc (node exit), cần xác định sao cho khi thực hiện thì độ đo kiểm thử tương ứng được thỏa mãn.
- **Bước 5: Thiết kế bộ dữ liệu kiểm thử:** Với tất cả các đường đi đã xác định phía trên, sinh ra một ca kiểm thử tương ứng cho mỗi đường.
- **Bước 6: Sinh ra các đầu vào (input) tương ứng cho từng ca kiểm thử** để đạt được độ đo kiểm thử tương ứng đã xác định.

- **Bước 7: Thực hiện kiểm thử, quan sát và đánh giá kết quả kiểm thử:** Chạy chương trình với các ca kiểm thử đã thiết kế, ghi nhận và quan sát các kết quả đầu ra của chương trình để kiểm tra xem chương trình có hoạt động đúng hay không, có phát sinh lỗi nào không, nhằm có những hành động phù hợp như tìm kiếm các lỗi liên quan.

2 Bài 2

2.1 Đồ thị dòng điều khiển cho hàm Grade ứng với độ đo C_1 và C_2



Hình 2: Đồ thị dòng điều khiển cho hàm **Grade**

2.2 Các đường đi và các ca kiểm thử với độ đo C_1

Để đảm bảo độ phủ độ đo C_1 đạt 100%, tất cả các câu lệnh trong chương trình đều được thực thi tối thiểu 1 lần sau khi chạy tất cả các ca kiểm thử.

STT	Path	Input	Output
1	0 1 2(T) 3 14	score = 100	I
2	0 1 2(F) 4(T) 5 13 14	score = 9	A
3	0 1 2(F) 4(F) 6(T) 7 13 14	score = 8	B
4	0 1 2(F) 5(F) 6(F) 8(T) 9 13 14	score = 7	C
5	0 1 2(F) 5(F) 6(F) 8(F) 10(T) 11 13 14	score = 6	D
6	0 1 2(F) 5(F) 6(F) 8(F) 10(F) 12 13 14	score = 2	F

Bảng 1: Các ca kiểm thử với độ đo C_1

Với bảng các ca kiểm thử như trên, $S_{cov} = 15/15 = 1$

2.3 Các đường đi và các ca kiểm thử với độ đo C_2

Để đảm bảo độ phủ độ đo C_2 đạt 100%, ta cần chắc chắn rằng mỗi nhánh của chương trình sinh ra từ các cấu trúc điều kiện được thực thi tối thiểu một lần.

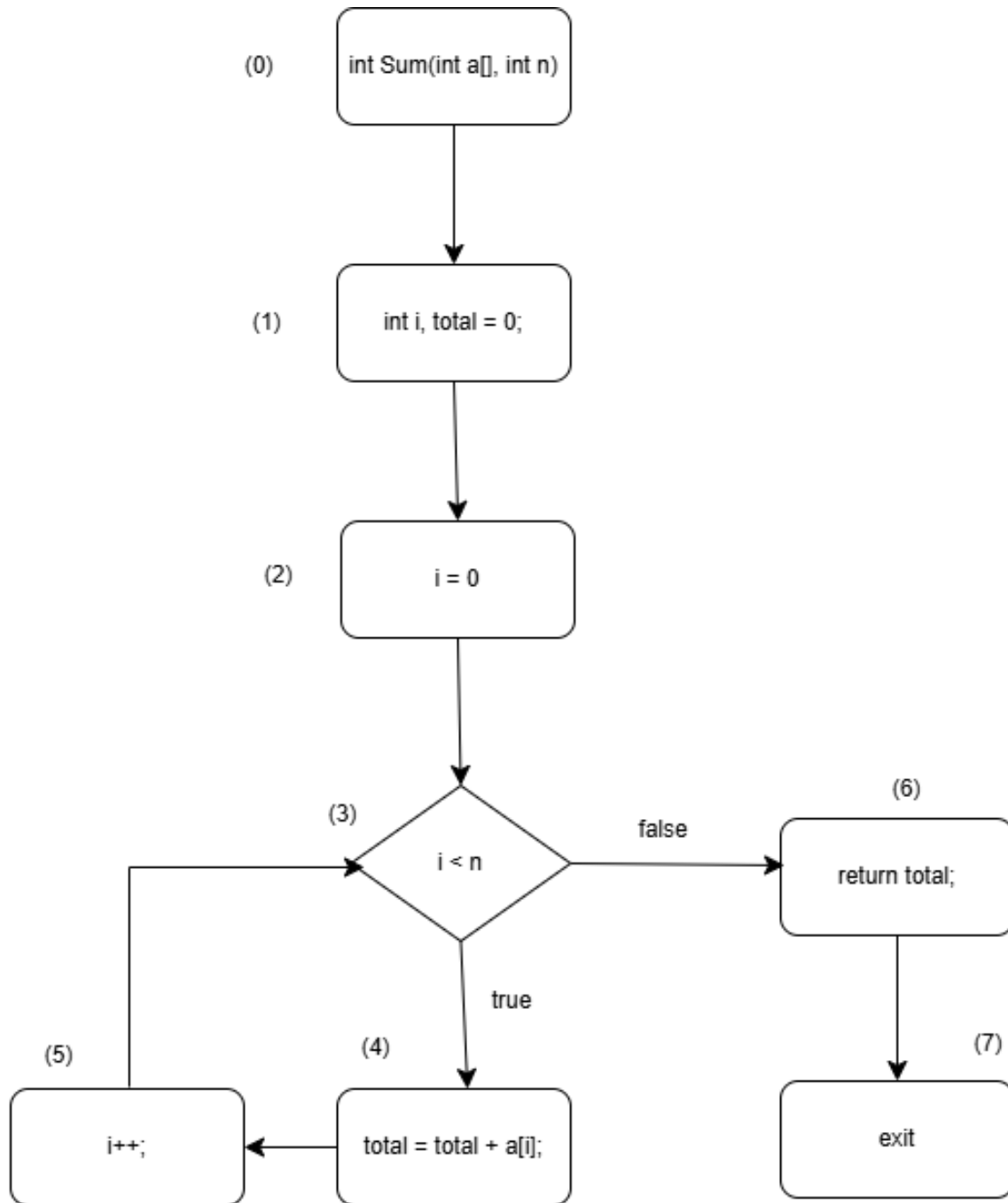
STT	Path	Input	Output
1	0 1 2(T) 3 14	score = 100	I
2	0 1 2(F) 4(T) 5 13 14	score = 9	A
3	0 1 2(F) 4(F) 6(T) 7 13 14	score = 8	B
4	0 1 2(F) 5(F) 6(F) 8(T) 9 13 14	score = 7	C
5	0 1 2(F) 5(F) 6(F) 8(F) 10(T) 11 13 14	score = 6	D
6	0 1 2(F) 5(F) 6(F) 8(F) 10(F) 12 13 14	score = 2	F

Bảng 2: Các ca kiểm thử với độ đo C_2

Với bảng các ca kiểm thử như trên, $B_{cov} = 10/10 = 1$

3 Bài 3

3.1 Đồ thị dòng điều khiển cho hàm Sum ứng với độ đo C_1 và C_2



Hình 3: Đồ thị dòng điều khiển cho hàm **Sum**

3.2 Các đường đi và các ca kiểm thử với độ đo C_1

Để đảm bảo độ phủ độ đo C_1 đạt 100%, tất cả các câu lệnh trong chương trình đều được thực thi tối thiểu 1 lần sau khi chạy tất cả các ca kiểm thử.

Với bảng các ca kiểm thử như trên, $S_{cov} = 8/8 = 1$

STT	Path	Input	Output
1	0 1 2 3(T) 4 5 3(F) 6 7	$a[] = 10, n = 1$	total = 10

Bảng 3: Các ca kiểm thử với độ đo C_1

3.3 Các đường đi và các ca kiểm thử với độ đo C_2

Để đảm bảo độ phủ độ đo C_2 đạt 100%, ta cần chắc chắn rằng mỗi nhánh của chương trình sinh ra từ các cấu trúc điều kiện được thực thi tối thiểu một lần.

STT	Path	Input	Output
1	0 1 2 3(F) 6 7	$a[] = [10, 20, 30], n = 0$	total = 0
2	0 1 2 3(T) 4 5 3(F) 6 7	$a[] = 10, n = 1$	total = 10

Bảng 4: Các ca kiểm thử với độ đo C_2

Với bảng các ca kiểm thử như trên, $B_{cov} = 2/2 = 1$

3.4 Các ca kiểm thử để kiểm thử vòng lặp for

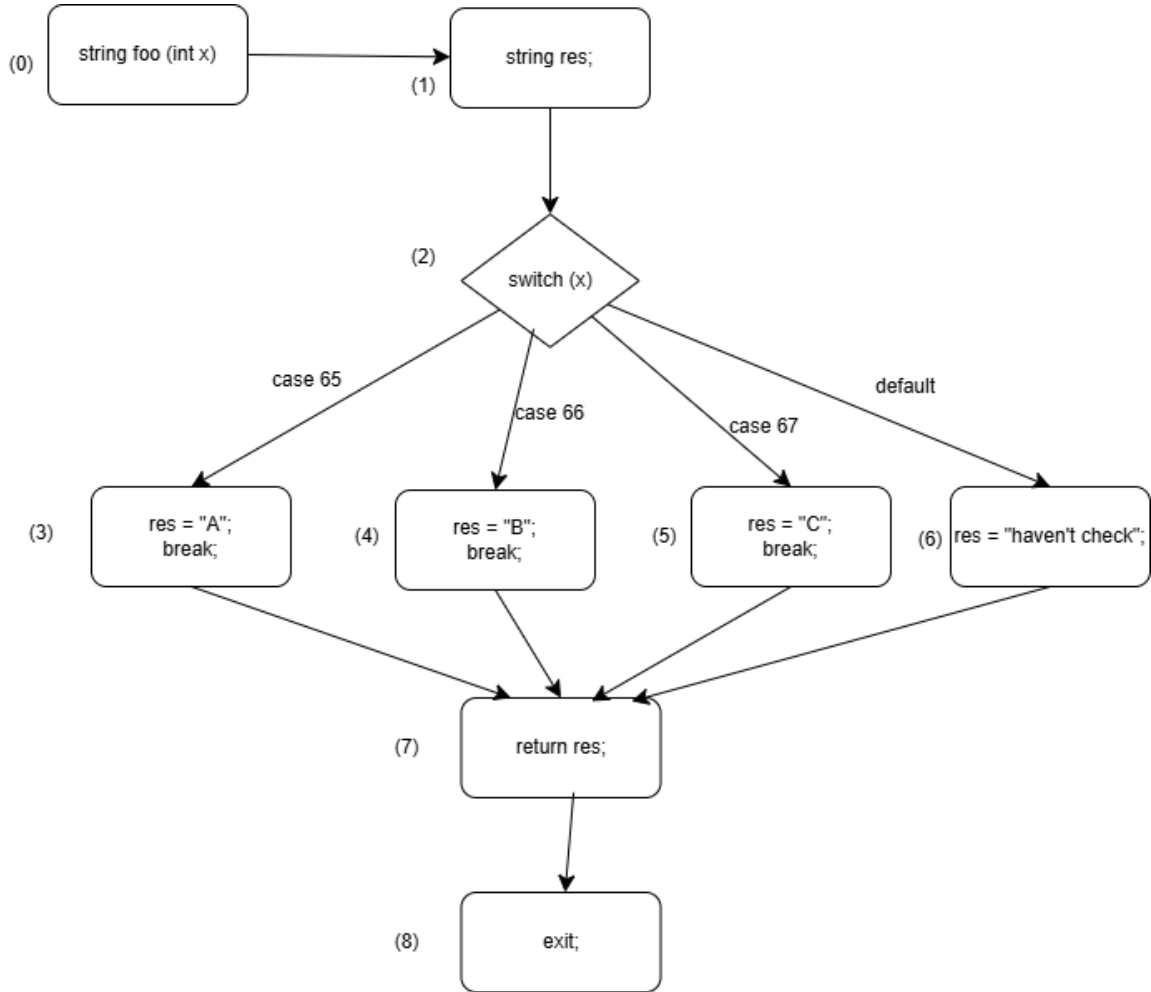
Trong trường hợp lặp $n + 1$ lần thì chương trình sẽ ném ngoại lệ hoặc trả lại lỗi vì truy xuất phần tử ở ngoài mảng.

STT	Số lần lặp	Input	Output
1	0	$a[] = [], n = 0$	total = 0
2	1	$a[] = [10], n = 1$	total = 10
3	2	$a[] = [10, 20], n = 2$	total = 30
4	3 ($k < n - 1$)	$a[] = [10, 20, 30], n = 3$	total = 60
5	$n - 1$	$a[] = [10, 20, 30, 40], n = 4$	total = 100
6	n	$a[] = [10, 20, 30, 40], n = 4$	total = 100
7	$n + 1$	$a[] = [10, 20, 30, 40], n = 5$	total = ? (không xác định)

Bảng 5: Các ca kiểm thử với vòng lặp **for**

4 Bài 4

4.1 Đồ thị dòng điều khiển cho hàm foo ứng với độ đo C_2



Hình 4: Đồ thị dòng điều khiển cho hàm **foo**

4.2 Các đường đi và các ca kiểm thử ứng với độ đo C_2

Để đảm bảo độ phủ độ đo C_2 đạt 100%, ta cần chắc chắn rằng mỗi nhánh của chương trình sinh ra từ các cấu trúc điều kiện được thực thi tối thiểu một lần.

STT	Path	Input	Output
1	0 1 2(default) 6 7 8	x = 100	res = "haven't check"
2	0 1 2(67) 5 7 8	x = 67	res = "C"
3	0 1 2(66) 4 7 8	x = 66	res = "B"
2	0 1 2(65) 5 7 8	x = 65	res = "A"

Bảng 6: Các ca kiểm thử với độ đo C_2

Với bảng các ca kiểm thử như trên, $B_{cov} = 4/4 = 1$

5 Bài 5

5.1 Mô tả bài toán

Một công ty thuộc lĩnh vực vận chuyển hàng hóa tính phí giao hàng dựa trên khoảng cách (x), trọng lượng của hàng hóa (y) và thời gian giao hàng mong muốn (z). Hãy tính tổng số tiền khách hàng phải trả cho phí vận chuyển, với các điều kiện sau:

- Với khoảng cách (km):
 - Dưới 10km: 5.000 VND/km
 - Từ 10km đến dưới 50km: 4.500 VND/km
 - Từ 50km trở lên: 4.000 VND/km
- Với trọng lượng của hàng hóa (kg):
 - Dưới 10kg: Không thu thêm phí
 - Từ 10kg trở lên: 50.000 VND
- Với thời gian giao hàng mong muốn (h):
 - Dưới 12h: Phụ phí 50.000 VND
 - Từ 12h trở lên: Không thu thêm phí

Đầu vào: Ba số x, y, z với $x, y \in R, z \in Z, 0 < x \leq 1000, 0 < y \leq 100, 0 < z \leq 24$
(x, y làm tròn đến 2 chữ số sau dấu thập phân)

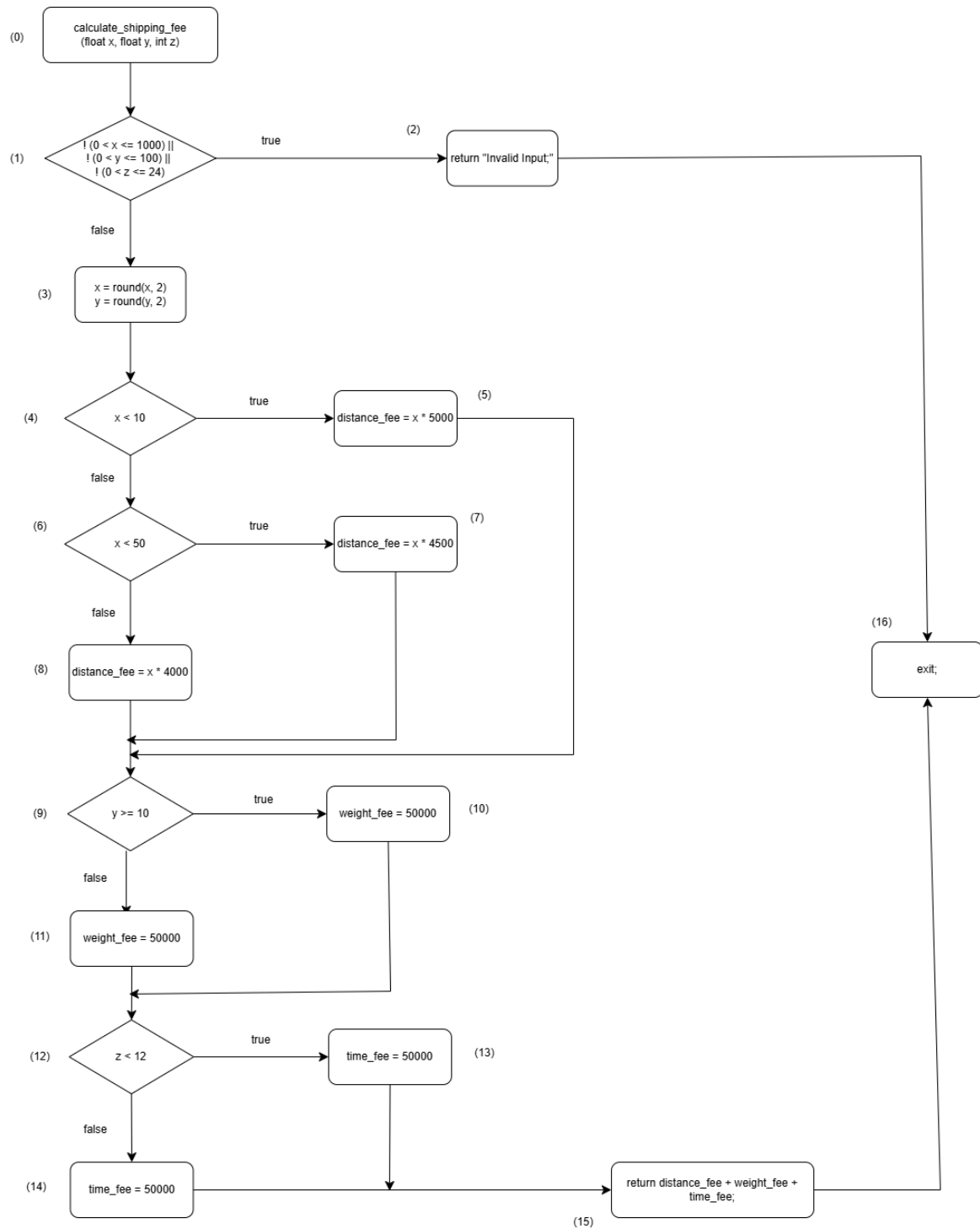
Đầu ra: Số tiền mà khách hàng phải trả.

Link đến mã nguồn trên Github: [Mã nguồn và các ca kiểm thử](#)

5.2 Chương trình của bài toán

```
1 def calculate_shipping_fee(x: float, y: float, z: int) -> float:
2     if not (0 < x <= 1000) or not (0 < y <= 100) or not (0 < z <= 24):
3         return "Invalid Input"
4
5     x, y = round(x, 2), round(y, 2)
6     distance_fee = x * 5000 if x < 10 else (x * 4500 if x < 50 else x * 4000)
7     weight_fee = 50000 if y >= 10 else 0
8     time_fee = 50000 if z < 12 else 0
9
10    return distance_fee + weight_fee + time_fee
```

5.3 Đồ thị dòng điều khiển của hàm `calculate_shipping_fee` tương ứng với độ phủ C2



Hình 5: Đồ thị dòng điều khiển cho hàm `calculate_shipping_fee`

5.4 Thiết kế các ca kiểm thử

Để đảm bảo độ phủ độ đo C_2 đạt 100%, ta cần chắc chắn rằng mỗi nhánh của chương trình sinh ra từ các cấu trúc điều kiện được thực thi tối thiểu một lần.

STT	Path	Input	Output
1	0 1(T) 2 16	$x = 1000.01, y = 16.12, z = 22$	Invalid Input
2	0 1(F) 3 4(T) 5 9(T) 10 12(T) 13 15 16	$x = 9.75, y = 12.25, z = 15$	98750
3	0 1(F) 3 4(F) 6(T) 7 9(F) 11 12(F) 14 15 16	$x = 42.5, y = 9.25, z = 20$	191250
4	0 1(F) 3 4(F) 6(F) 8 9(F) 11 12(T) 13 15 16	$x = 100.01, y = 6.45, z = 10$	450040

Bảng 7: Các ca kiểm thử với độ đo C_2

Với bảng các ca kiểm thử như trên, $B_{cov} = 10/10 = 1$

5.5 Kết quả khi kiểm thử

STT	Input	Expected Output	Actual Output	Result
1	$x = 1000.01, y = 16.12, z = 22$	Invalid Input	Invalid Input	Pass
2	$x = 9.75, y = 12.25, z = 15$	98750	98750	Pass
3	$x = 42.5, y = 9.25, z = 20$	191250	191250	Pass
4	$x = 100.01, y = 6.45, z = 10$	450040	450040	Pass

Bảng 8: Kết quả khi kiểm thử với bộ test case