

1:

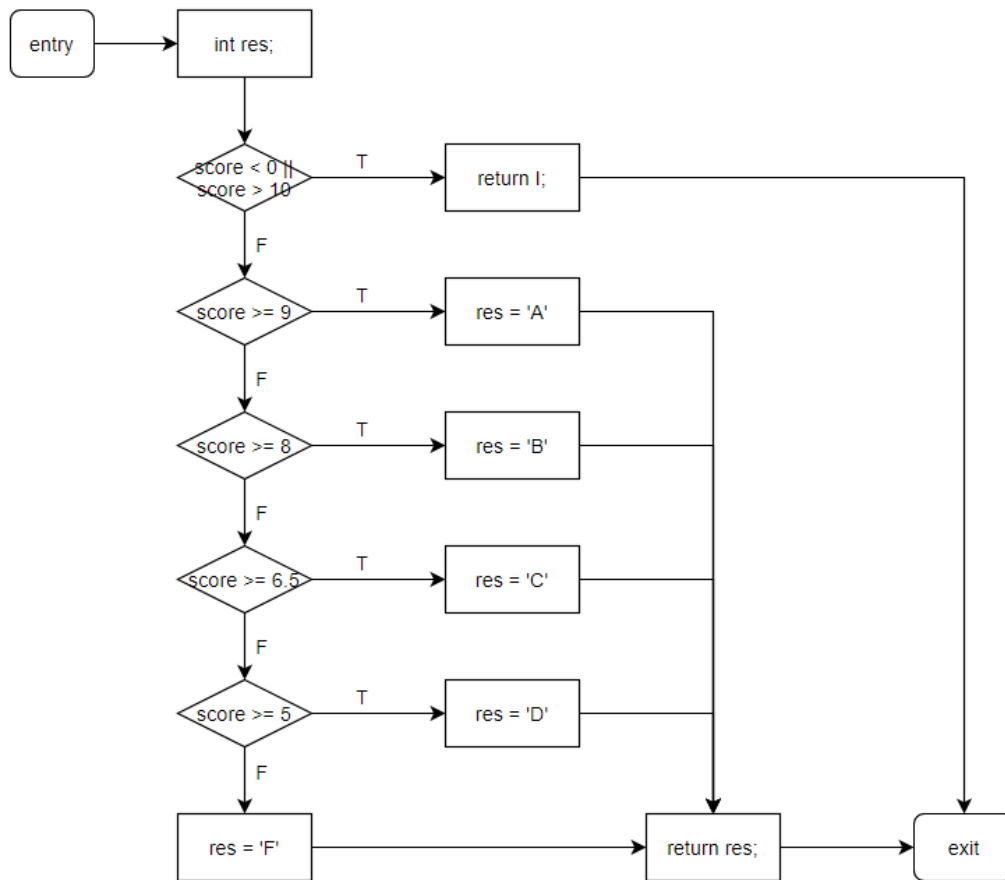
9. Cho hàm được viết bằng ngôn ngữ C như Đoạn mã 6.1.

Đoạn mã 6.1: Mã nguồn của hàm Grade

```
char Grade(int score){  
  
    int res;  
    if(score < 0 || score > 10)  
        return 'I';  
    if(score >= 9)  
        res = 'A';  
    else  
        if(score >= 8)  
            res = 'B';  
        else  
            if(score >= 6.5)  
                res = 'C';  
            else  
                if(score >= 5)  
                    res = 'D';  
                else  
                    res = 'F';  
  
    return res;  
}
```

- Hãy xây dựng đồ thị dòng điều khiển cho hàm Grade ứng với độ đo  $C_1$  và  $C_2$ .
- Hãy sinh các đường đi và các ca kiểm thử với độ đo  $C_1$ .
- Hãy sinh các đường đi và các ca kiểm thử với độ đo  $C_2$ .

## CFG



```
1: int res
2: if (score < 0 || score > 10)
3: return I
4: if (score >= 9)
5: res = 'A'
6: if (score >= 8)
7: res = 'B'
8: if (score >= 6.5)
9: res = 'C'
10: if (score >= 5)
11: res = 'D'
12: res = 'F'
13: return res
```

Độ đo C1:

Input: score	Path	Output
11	1 -> 2(T) -> 3	I
9	1 -> 2(F) -> 4(T) -> 5 -> 13	'A'
8	1 -> 2(F) -> 4(F) -> 6(T) -> 7 -> 13	'B'
7	1 -> 2(F) -> 4(F) -> 6(F) -> 8(T) -> 9 -> 13	'C'
5	1 -> 2(F) -> 4(F) -> 6(F) -> 8(F) -> 10(T) -> 11 -> 13	'D'
4	1 -> 2(F) -> 4(F) -> 6(F) -> 8(F) -> 10(F) -> 12 -> 13	'F'

Độ đo C2:

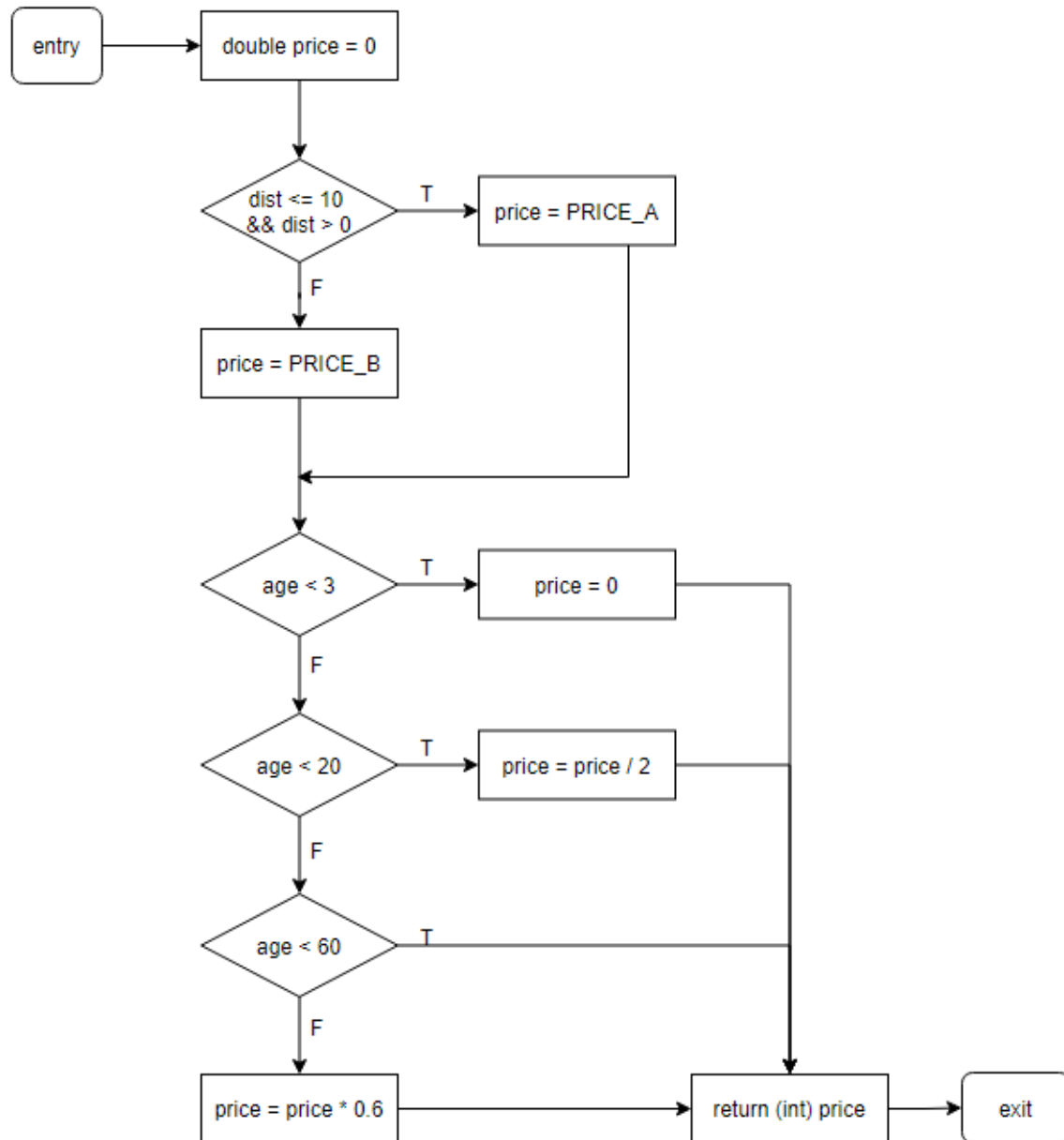
Input: score	Path	Output
-1	1 -> 2(T) -> 3	I
10	1 -> 2(F) -> 4(T) -> 5 -> 13	'A'
8	1 -> 2(F) -> 4(F) -> 6(T) -> 7 -> 13	'B'
7	1 -> 2(F) -> 4(F) -> 6(F) -> 8(T) -> 9 -> 13	'C'
6	1 -> 2(F) -> 4(F) -> 6(F) -> 8(F) -> 10(T) -> 11 -> 13	'D'
3	1 -> 2(F) -> 4(F) -> 6(F) -> 8(F) -> 10(F) -> 12 -> 13	'F'

2:

<https://github.com/huuduy16/testing-homework>

(code này có bug)

CFG



PRICE\_A = 100

PRICE\_B = 150

1: double price = 0

2: if (dist <= 10 && dist > 0)

3: price = PRICE\_A

4: price = PRICE\_B

5: if (age < 3)

6: price = 0

7: if (age < 20)

8: price = price / 2

9: if (age < 60)

10: price = price \* 0.6

11: return (int) price

Độ đo C2:

Input: (age, dist)	Path	Output
(1, 5)	1 -> 2(T) -> 3 -> 5(T) -> 6 -> 11	0
(10, 20)	1 -> 2(F) -> 4 -> 5(F) -> 7(T) -> 8 -> 11	75
(50, 5)	1 -> 2(T) -> 3 -> 5(F) -> 7(F) -> 9(T) -> 11	100
(100, 5)	1 -> 2(T) -> 3 -> 5(F) -> 7(F) -> 9(F) -> 10 -> 11	60