

# BÁO CÁO KIỂM THỬ PHẦN MỀM – KIỂM THỬ DÒNG ĐIỀU KHIỂN

---

Họ và tên: Hoàng Quốc Bảo

MSSV: 23020012

## BÀI 1 – PHÂN TÍCH VÀ KIỂM THỬ HÀM GRADE

Đề bài yêu cầu phân tích và kiểm thử hàm Grade được viết bằng ngôn ngữ C. Hàm Grade nhận đầu vào là điểm số (score) và trả về ký tự thể hiện mức độ đánh giá tương ứng. Mã nguồn hàm Grade như sau:

```
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; }
```

### 1. Phân tích cấu trúc điều khiển

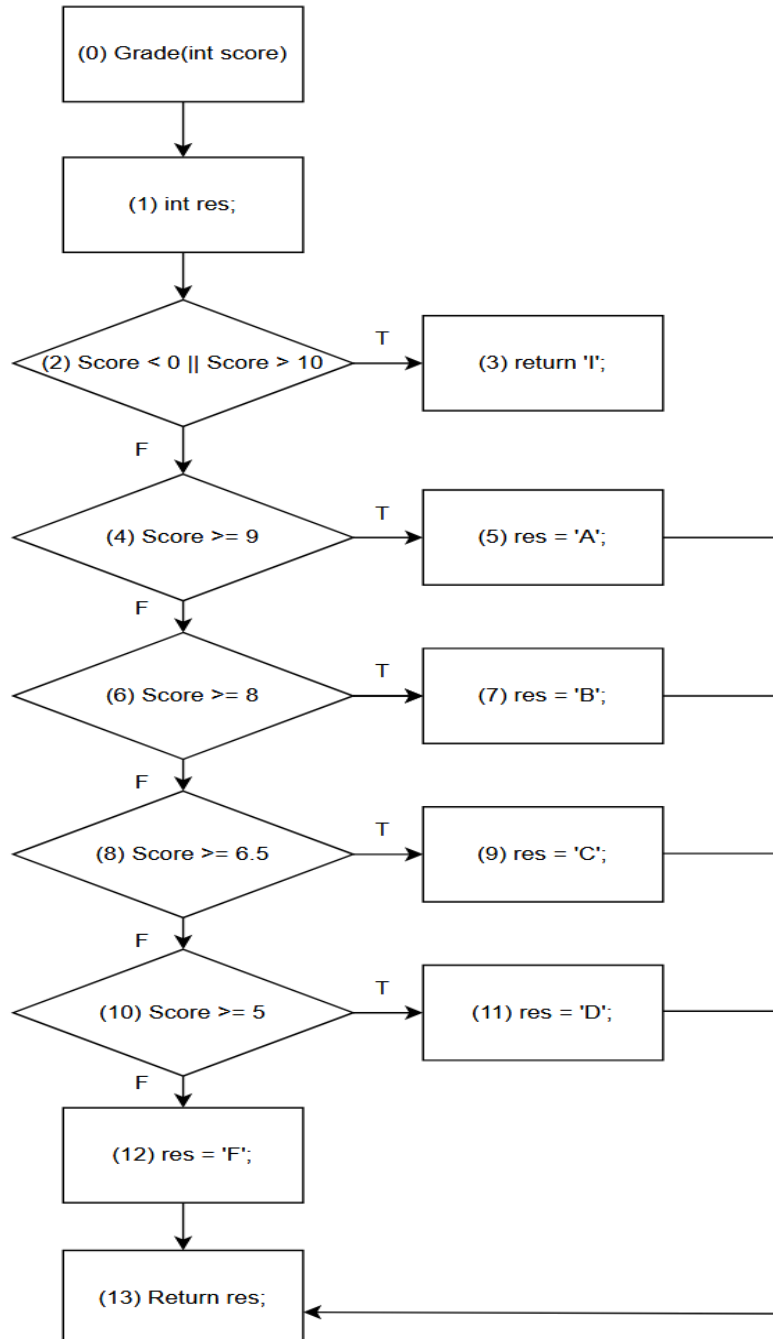
Hàm Grade bao gồm một chuỗi các câu lệnh điều kiện lồng nhau (if-else). Có tổng cộng 5 điều kiện được kiểm tra theo thứ tự mức điểm.

Các nhánh điều kiện:

- if(score < 0 || score > 10): trả về 'I' (Invalid)
- if(score >= 9): trả về 'A'
- else if(score >= 8): trả về 'B'
- else if(score >= 6.5): trả về 'C'
- else if(score >= 5): trả về 'D'
- else: trả về 'F'

## 2. Xây dựng đồ thị dòng điều khiển

Đồ thị dòng điều khiển (Control Flow Graph - CFG) mô tả các đường đi có thể trong hàm Grade. Mỗi điều kiện tương ứng với một nút rẽ nhánh, các kết quả trả về tương ứng với các nút kết thúc.



## 3. Độ đo C1 (Độ bao phủ câu lệnh)

Độ đo C1 yêu cầu mỗi câu lệnh trong chương trình phải được thực thi ít nhất một lần. Ta cần chọn các giá trị score sao cho mọi nhánh đều được đi qua.

Các đường đi, test case tương ứng với C1:

1. đường: 0-1-2(T)-3.	T(score = -1)
2. đường: 0-1-2(F)-4(T)-5-13.	T(score = 9)
3. đường: 0-1-2(F)-4(F)-6(T)-7-13.	T(score = 8)
4. đường: 0-1-2(F)-4(F)-6(F)-8(T)-9-13.	T(score = 7)
5. đường: 0-1-2(F)-4(F)-6(F)-8(F)-10(T)-11-13.	T(score = 5)
6. đường: 0-1-2(F)-4(F)-6(F)-8(F)-10(F)-12-13.	T(score = 4)

#### 4. Độ đo C2 (Độ bao phủ nhánh)

Độ đo C2 yêu cầu mỗi nhánh (true/false) của các câu điều kiện phải được thực hiện ít nhất một lần. Từ đó ta cần chọn các giá trị đảm bảo cả hai phía của mỗi điều kiện đều được xét.

Các đường đi, test case tương ứng với C2:

1. đường: 0-1-2(T)-3.	T(score = -1)
2. đường: 0-1-2(F)-4(T)-5-13.	T(score = 9)
3. đường: 0-1-2(F)-4(F)-6(T)-7-13.	T(score = 8)
4. đường: 0-1-2(F)-4(F)-6(F)-8(T)-9-13.	T(score = 7)
5. đường: 0-1-2(F)-4(F)-6(F)-8(F)-10(T)-11-13.	T(score = 5)
6. đường: 0-1-2(F)-4(F)-6(F)-8(F)-10(F)-12-13.	T(score = 4)

#### 5. Kết luận

Qua quá trình phân tích, ta đã xác định được các đường đi kiểm thử và ca kiểm thử tương ứng cho hai độ đo bao phủ câu lệnh (C1) và bao phủ nhánh (C2). Hàm Grade có cấu trúc điều khiển dạng cây if-else lồng nhau, phù hợp cho việc áp dụng kiểm thử hộp trắng.

## BÀI 2 – PHÂN TÍCH VÀ KIỂM THỬ HÀM SUM

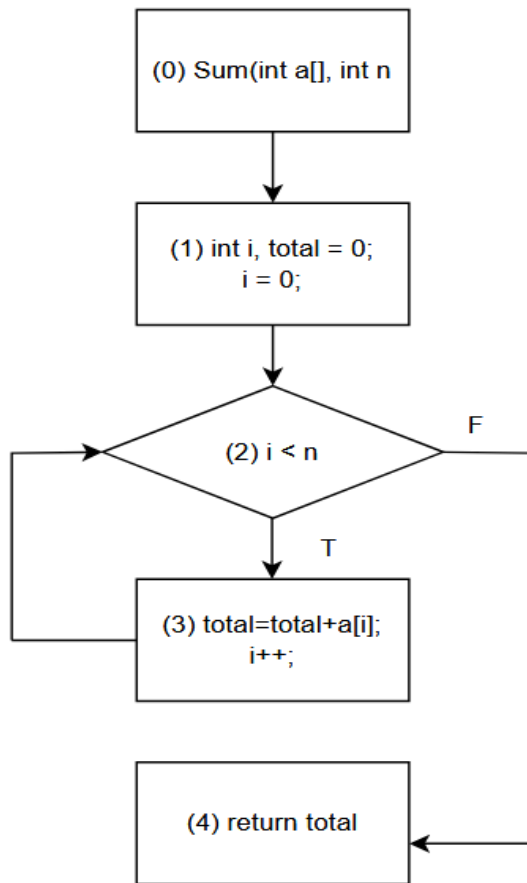
Hàm Sum tính tổng các phần tử trong mảng a[] có n phần tử. Yêu cầu bài toán là xây dựng đồ thị dòng điều khiển, sinh các đường đi và ca kiểm thử theo độ đo C1, C2, và kiểm thử vòng lặp for.

```
int Sum(int a[], int n){  
    int i, total = 0;  
    for(i=0; i<n; i++)  
        total = total + a[i];  
    return total;  
}
```

### 1. Phân tích dòng điều khiển

Hàm bao gồm một vòng lặp for duy nhất, trong đó điều kiện dừng là  $i < n$ . Mỗi vòng lặp cộng dồn giá trị của  $a[i]$  vào total. Sau khi vòng lặp kết thúc, hàm trả về total.

## 2. Đồ thị dòng điều khiển



## 3. Độ đo C1 (Độ bao phủ câu lệnh)

Độ đo C1 yêu cầu mỗi câu lệnh được thực thi ít nhất một lần. Ta cần các trường hợp:

- Trường hợp không vào vòng lặp:  $n = 0$ .
- Trường hợp có ít nhất một lần lặp:  $n > 0$ .

Các đường đi, test case tương ứng với C1:

1. đường: 0-1-2-3-4	T(a={1}, n = 1)
---------------------	-----------------

## 4. Độ đo C2 (Độ bao phủ nhánh)

Độ đo C2 yêu cầu mỗi nhánh (true/false) của điều kiện trong vòng lặp phải được thực hiện ít nhất một lần.

Điều kiện duy nhất:  $i < n$ .

Cần có ít nhất một lần  $i < n$  (true) và một lần  $i \geq n$  (false).

Các đường đi, test case tương ứng với C2:

1. đường: 0-1-2(T)-3-2(F)-4	T(a={1}, n = 1)
-----------------------------	-----------------

## 5. Kiểm thử vòng lặp for

Để kiểm thử vòng lặp for, áp dụng ba ca chính:

- Không thực hiện vòng lặp ( $n = 0$ )  $T(a=\{\}, n = 0)$
- Thực hiện 1 lần ( $n = 1$ )  $T(a=\{1\}, n = 1)$
- Thực hiện 2 lần ( $n = 2$ )  $T(a=\{1, 2\}, n = 2)$
- Thực hiện k lần ( $n = 4$ ) ( $2 < k < n - 1$ )  $T(a=\{1, 2, 3, 4\}, n = 4)$
- Thực hiện n - 1 lần : Không tồn tại
- Thực hiện n lần : Không tồn tại
- Thực hiện n + 1 lần : Không tồn tại

→ Đảm bảo tính đúng đắn và bao phủ các nhánh vòng lặp.

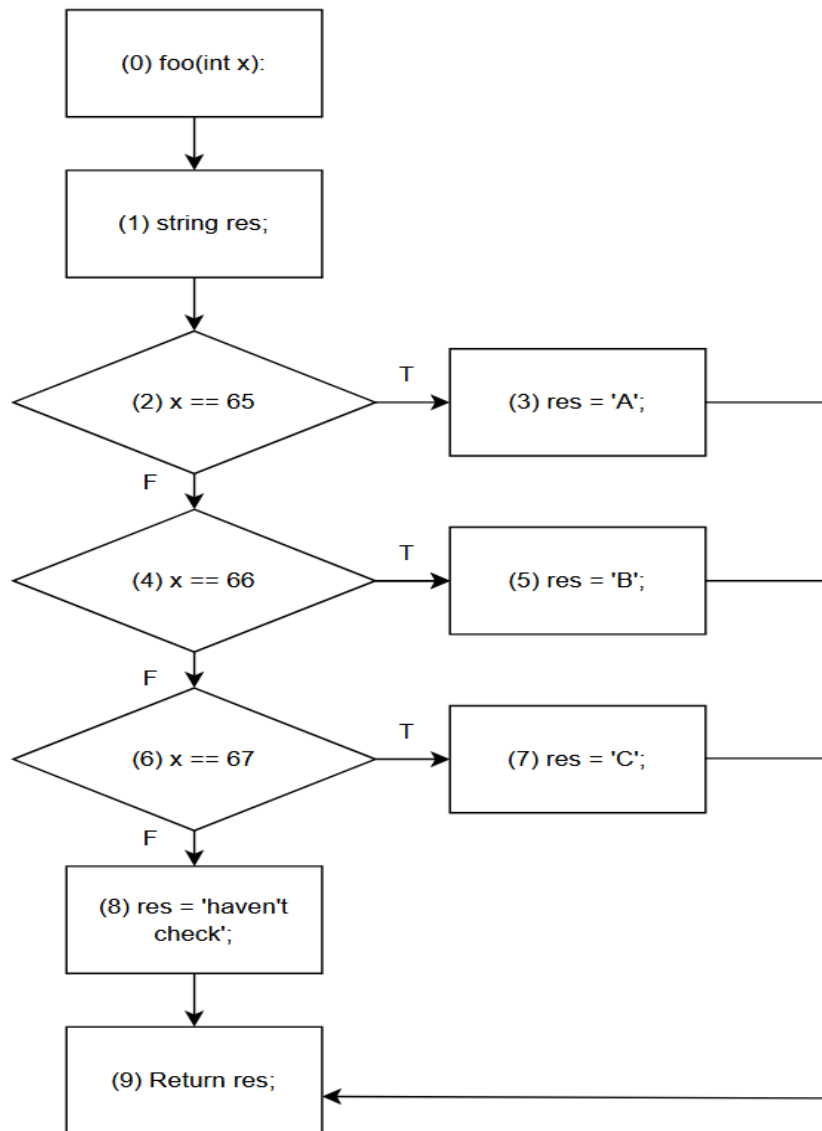
## BÀI 3 – PHÂN TÍCH VÀ KIỂM THỬ HÀM FOO

```
string foo(int x){  
    string res;  
    switch(x){  
        case 65: res = "A"; break;  
        case 66: res = "B"; break;  
        case 67: res = "C"; break;  
        default: res = "haven't check";  
    }  
    return res;  
}
```

### 1. Phân tích dòng điều khiển

Hàm foo sử dụng cấu trúc switch-case để gán giá trị cho biến res tùy thuộc vào giá trị của x. Có 3 nhánh case và 1 nhánh default.

## 2. Đồ thị dòng điều khiển



## 3. Độ đo C2 (bao phủ nhánh)

Mỗi case trong switch tương ứng với một nhánh điều khiển. Để đạt bao phủ C2, phải có ít nhất một ca kiểm thử cho mỗi case và một ca cho default.

Các đường đi, test case tương ứng với C2:

1. đường: 0-1-2(T)-3-9.	T(65)
2. đường: 0-1-2(F)-4(T)-5-9.	T(66)
3. đường: 0-1-2(F)-4(F)-6(T)-7-9.	T(67)
4. đường: 0-1-2(F)-4(F)-6(F)-8-9.	T(60)

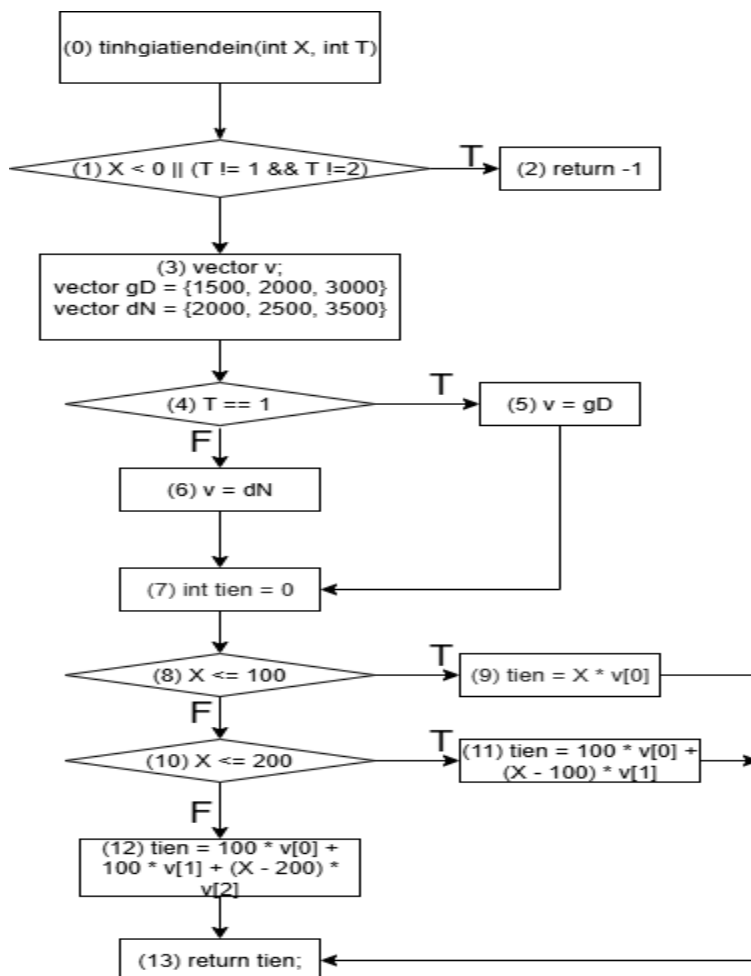
## BÀI 4 – PHÂN TÍCH VÀ KIỂM THỬ HÀM TINHGIATIENDIEN

```
int tinhGiaTienDien(int X, int T) {  
    if ((X < 0) || (T != 1 && T != 2)) return -1;  
    vector<int> v = T == 1 ? giaDinh : doanhNghiep;  
    int tien = 0;  
    if (X <= 100) tien = X * v[0];  
    else if (X <= 200) tien = 100 * v[0] + (X - 100) * v[1];  
    else tien = 100 * v[0] + 100 * v[1] + (X - 200) * v[2];  
    return tien;  
}
```

### 1. Phân tích yêu cầu và cấu trúc điều khiển

Hàm tính giá điện dựa trên số điện X và loại khách hàng T. Nếu dữ liệu không hợp lệ, trả về -1. Sau đó, chọn bảng giá phù hợp và áp dụng tính tiền theo các mức bậc thang.

### 2. Đồ thị dòng điều khiển





#### 4. Độ đo C2

Các nhánh kiểm tra bao gồm:

- Điều kiện đầu tiên ( $X < 0$  hoặc T sai)
- $X \leq 100$
- $X \leq 200$

Các ca kiểm thử đảm bảo mỗi nhánh true/false đều được xét.

Các đường đi, test case tương ứng với C2:

1. đường: 0-1(T)-2.	T(X = -1, T = 3)
2. đường: 0-1(F)-3-4(T)-5-7-8(T)-9-13.	T(X = 88, T = 1)
3. đường: 0-1-2(F)-3-4(F)-6-7-8(F)-10(T)-11-13.	T(X = 121, T = 2)
4. đường: 0-1-2(F)-3-4(F)-6-7-8(F)-10(F)-12-13.	T(X = 300, T = 2)

#### 5. Kết luận

Hàm có cấu trúc điều kiện lồng nhau với 3 mức giá điện và kiểm tra dữ liệu đầu vào. Các ca kiểm thử đã bao phủ toàn bộ nhánh và câu lệnh theo C1 và C2.

Link Github: <https://github.com/bao2811/testing.git>