

Họ và tên: Nguyễn Đức Tùng
MSV: 21020556

Bài tập dòng điều khiển

Code: <https://github.com/tung1883/kiemthu>

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

- Các bước kiểm thử theo phương pháp kiểm thử dòng điều khiển như sau:

1. Xây dựng đồ thị điều khiển của chương trình, tương ứng với độ đo kiểm thử đã cho
2. Tìm tất cả các đường đi trong đồ thị điều khiển thoả mãn yêu cầu của độ đo kiểm thử.
3. Với mỗi đường đi tìm được, thiết kế các ca kiểm thử
4. Thực hiện các ca kiểm thử và phân tích kết quả nhận được

Bài 2:

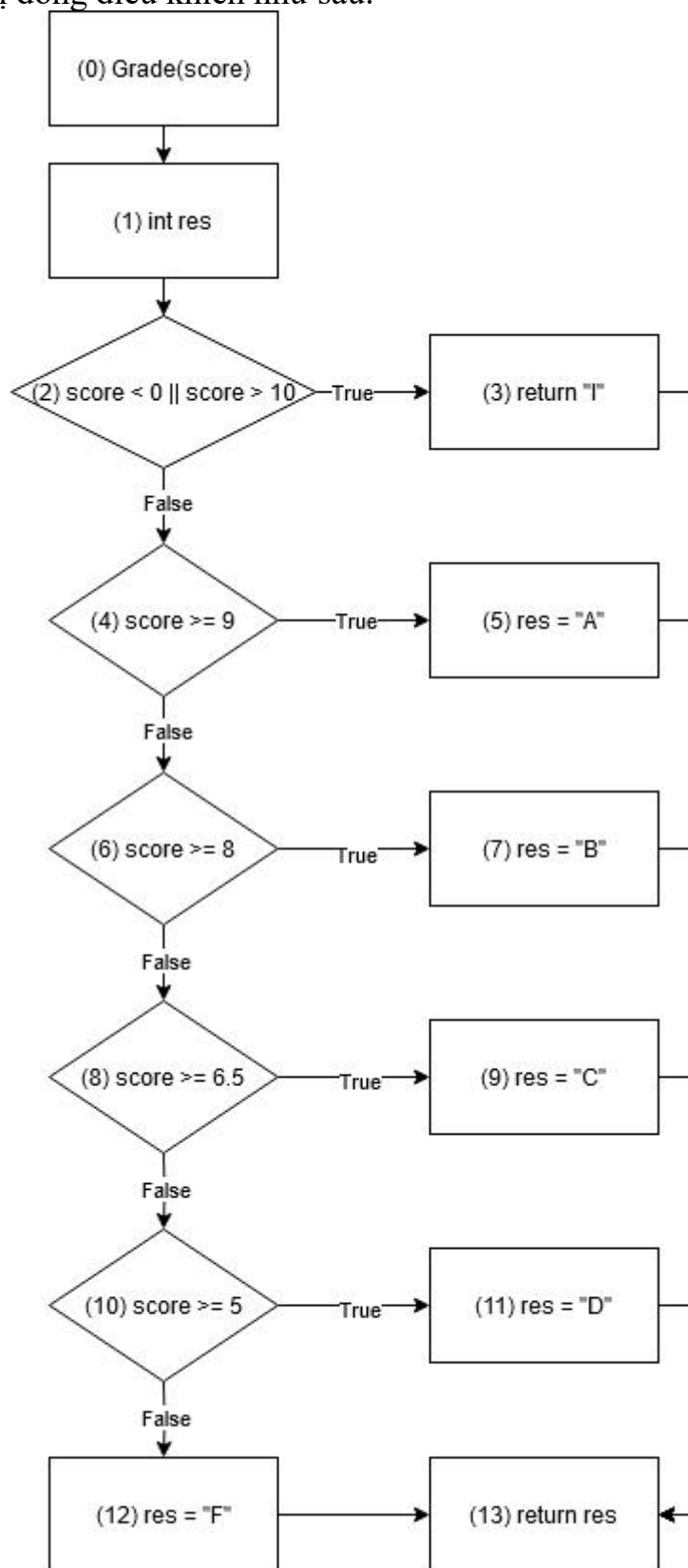
9. Cho hàm được viết bằng ngôn ngữ C như Đoạn mã 6.1.

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 .

- Ta có đồ thị dòng điều khiển như sau:



- Các ca kiểm thử với độ đo C1:

STT	Đường đi	Ca kiểm thử
1	0 - 1 - 2(T) - 3 - 13	score = -1
2	0 - 1 - 2(F) - 4(T) - 5 - 13	score = 9
3	0 - 1 - 2(F) - 4(F) - 6(T) - 7 - 13	score = 8
4	0 - 1 - 2(F) - 4(F) - 6(F) - 8(T) - 9 - 13	score = 6.5
5	0 - 1 - 2(F) - 4(F) - 6(F) - 8(F) - 10(T) - 11 - 13	score = 5
6	0 - 1 - 2(F) - 4(F) - 6(F) - 8(F) - 10(F) - 12 - 13	score = 4

- Ta thấy các ca kiểm thử trên có đã bao phủ hoàn toàn các nhánh => Đạt được độ bao phủ C2

Bài 3:

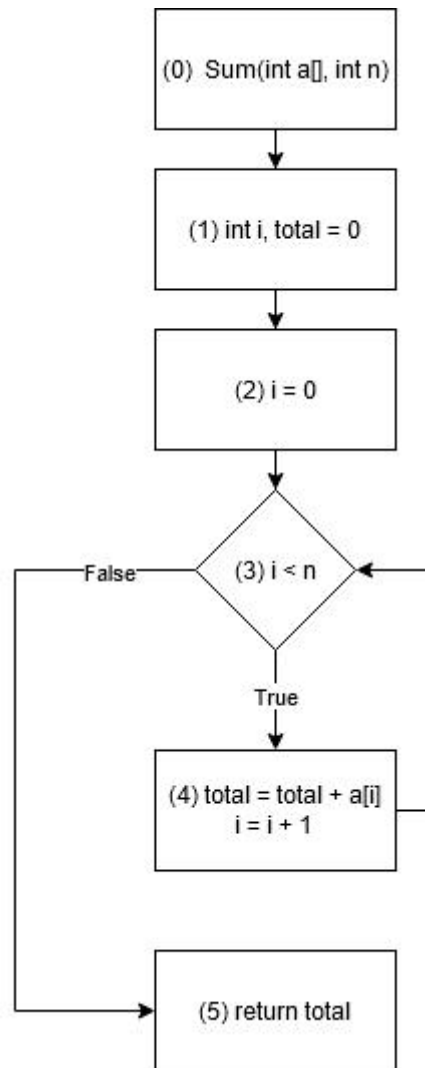
13. Cho hàm được viết bằng ngôn ngữ C như Đoạn mã 6.5.

Đoạn mã 6.5: Mã nguồn của hàm Sum

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

- Hãy xây dựng đồ thị dòng điều khiển cho hàm Sum 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 .
- Hãy sinh các ca kiểm thử để kiểm thử vòng lặp for.

- Ta có đồ thị dòng điều khiển như sau:



- Các ca kiểm thử với độ đo C1:

STT	Đường đi	Cá kiểm thử
1	0 - 1 - 2 - 3(T) - 4 - 3(F) - 5	$a[] = \{ 1 \}, n = 1$

- Các ca kiểm thử với độ đo C2:

STT	Đường đi	Cá kiểm thử
1	0 - 1 - 2 - 3(F) - 5	$a[] = \{ \}, n = 0$
2	0 - 1 - 2 - 3(T) - 4 - 3(T) - 4 - 3(F) - 5	$a[] = \{ 1, 2 \}, n = 2$

- Các ca kiểm thử vòng lặp for với N là số vòng lặp tối đa:

STT	Đường đi	Cá kiểm thử
1	0 - 1 - 2 - 3(F) - 5	$a[] = \{ \}, n = 0$
2	0 - 1 - 2 - 3(T) - 4 - 3(T) - 4 - 3(F) - 5	$a[] = \{ 1 \}, n = 1$
3	0 - 1 - 2 - 3(T) - 4 - 3(T) - 4 - 3(F) - 5	$a[] = \{ 1, 2 \}, n = 2$

	...	
N - 1		$a[] = \{ 1, \dots, N - 1 \}, n = N - 1$
N		$a[] = \{ 1, \dots, N \}, n = N$
N + 1		$a[] = \{ 1, \dots, N + 1 \}, n = N + 1$

Bài 4:

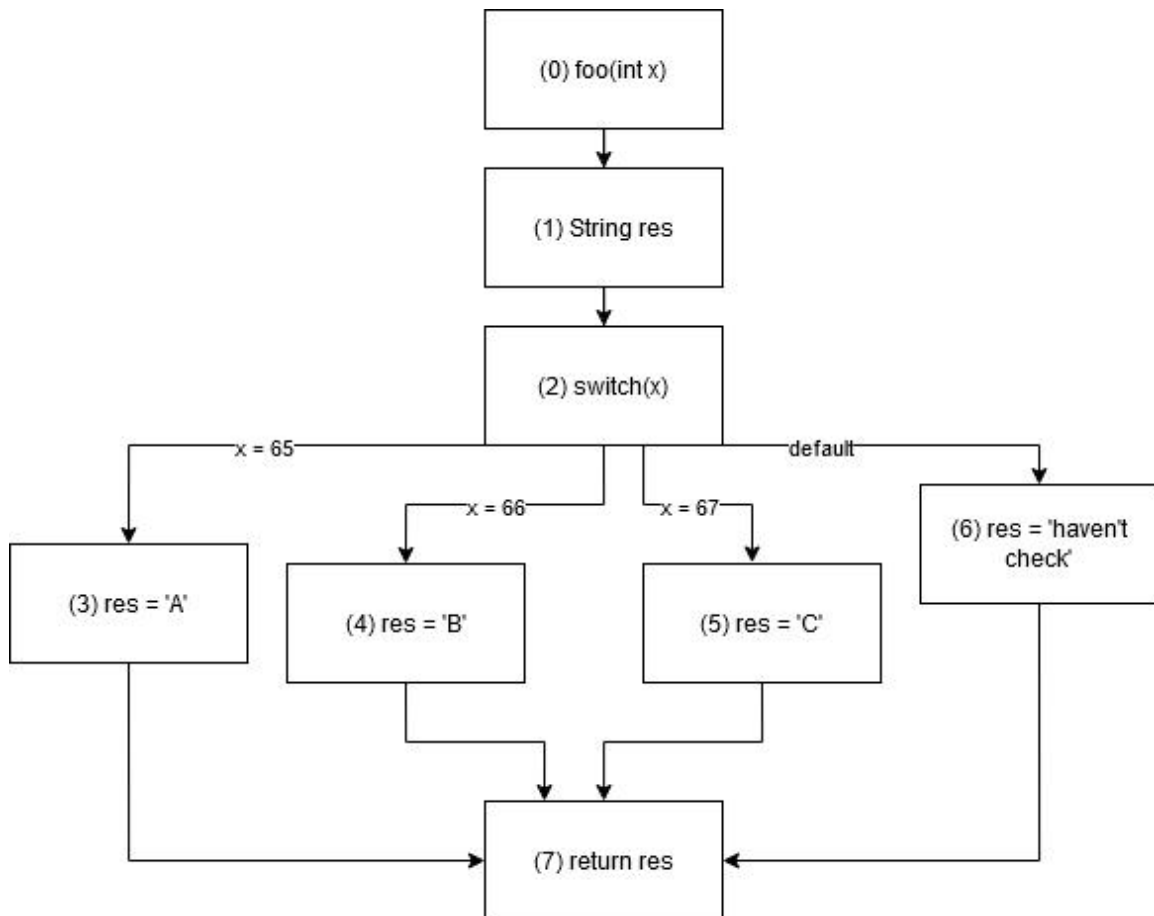
```

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;
}

```

- Xây dựng đồ thị dòng điều khiển cho hàm foo ứng với độ đo C2
- Hãy sinh các đường đi và các ca kiểm thử ứng với độ đo C2

- Ta có đồ thị dòng điều khiển như sau:



- Các ca kiểm thử với độ phủ C2:

STT	Đường đi	Ca kiểm thử
1	0 - 1 - 2 - 3 - 7	x = 65
2	0 - 1 - 2 - 4 - 7	x = 66
3	0 - 1 - 2 - 5 - 7	x = 67
4	0 - 1 - 2 - 6 - 7	x = 100

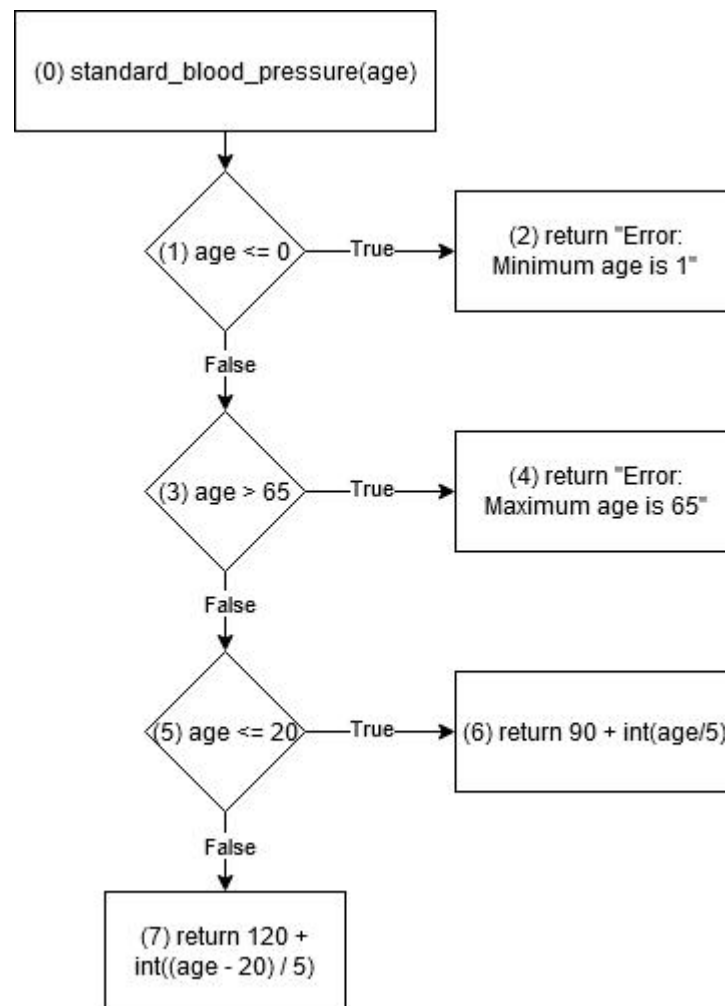
Bài 5:

Đề bài: Báo cáo phân tích, thiết kế các ca kiểm thử, và kiểm thử chương trình của bạn với độ phủ C2

+) Chương trình tính chỉ số huyết áp tiêu chuẩn với input là số tuổi của 1 người:

```
def standard_blood_pressure(age):
    if age <= 0:
        return "Error: Minimum age is 1"
    if age > 65:
        return "Error: Maximum age is 65"
    if age <= 20:
        return 90 + int(age / 5)
    return 120 + int((age - 20) / 5)
```

+) Đồ thị dòng điều khiển của chương trình như sau:



+) Các ca kiểm thử với độ phủ C2:

STT	Đường đi	Ca kiểm thử
1	0 - 1(T) - 2	x = -1

2	0 - 1(F) - 3(T) - 4	x = 67
3	0 - 1(F) - 3(F) - 5(T) - 6	x = 15
4	0 - 1(F) - 3(F) - 5(F) - 7	x = 21

+) Viết code kiểm thử với unittest của Python:

```
class ControlFlowTesting(unittest.TestCase):
    def test_1(self):
        self.assertEqual(standard_blood_pressure(-1),
                         "Error: Minimum age is 1")

    def test_2(self):
        self.assertEqual(standard_blood_pressure(67),
                         "Error: Maximum age is 65")

    def test_3(self):
        self.assertEqual(standard_blood_pressure(15), 93)

    def test_4(self):
        self.assertEqual(standard_blood_pressure(21), 120)

if __name__ == '__main__':
    unittest.main()
```

+) Kết quả sau khi chạy code kiểm thử:

```
....
-----
Ran 4 tests in 0.000s

OK
```