

## I. VERIFICATION AND VALIDATION

**Description:** The purpose is to help users to solve a 2-degree equation ( $ax^2+bx+c$ ).

**Spec:** Given input of  $a$ ,  $b$ , and  $c$ ; the system returns the outputs of  $x_1$  and  $x_2$  (extreme cases are temporarily not considered)

Two systems are developed as follows.

SYSTEM 1	SYSTEM 2
<div><div>a</div><div>b</div><div>c</div><div><div>Press here to get solutions</div><div><div><math>x_1</math></div><div><math>x_2</math></div></div></div></div>	<div><div>Step 1: DELTA calculation</div><div><div>a</div><div>b</div><div>c</div><div><div>Press here to get DELTA (discriminant)</div><div><math>\Delta</math></div></div></div></div> <div><div>Step 2: Solutions resolving</div><div><div>DELTA</div><div><div>Press here to get solutions</div><div><div><math>x_1</math></div><div><math>x_2</math></div></div></div></div></div>
<div>Code: <math>x_1 = \frac{-b + \sqrt{\Delta}}{2a}</math> <math>x_2 = \frac{-b - \sqrt{\Delta}}{2a}</math></div>	<div>Code: <math>\Delta = (b^2 - 4ac)</math> <math>x_1 = \frac{-b + \sqrt{\Delta}}{2a}</math> <math>x_2 = \frac{-b - \sqrt{\Delta}}{2a}</math></div>

What are the problems of those two systems? Write down your answer here.

Vì extreme cases are temporarily not considered nên chỉ xét trường hợp chuẩn ( $a \neq 0, \Delta \geq 0$ )

### 1. Lỗi của System 1:

#### Verification:

- Thiếu Delta
- Sai công thức nghiệm của  $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$ : Code thiếu dấu  $*$  ở  $2a$   
Công thức đúng:  $x_1 = \frac{-b + \sqrt{\Delta}}{2*a}$
- Sai công thức nghiệm của  $x_2 = \frac{-b - \sqrt{\Delta}}{2a}$ : Bỏ ngoặc sai, thiếu dấu  $*$  ở  $2a$   
Công thức đúng:  $x_2 = \frac{-b - \sqrt{\Delta}}{2*a}$

#### Validation:

Input:  $a = 1; b = 2; c = 1$

Expect:  $\Delta = 0; x_1 = -1; x_2 = -1$

Output: thiếu DELTA

### 2. Lỗi của System 2:

#### Verification:

- Sai công thức nghiệm của  $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$ : Code thiếu dấu  $*$  ở  $2a$   
Công thức đúng:  $x_1 = \frac{-b + \sqrt{\Delta}}{2*a}$

- Sai công thức nghiệm của  $x_2 = (-b - \sqrt{\text{DELTA}})/2a$ : Code thiếu dấu  $*$  ở  $2a$   
Công thức đúng:  $x_2 = (-b - \sqrt{\text{DELTA}}) / 2*a$

### Validation:

Input:  $a = 1$ ;  $b = 2$ ;  $c = 1$

Expect:  $\text{DELTA} = 0$ ;  $x_1 = -1$ ;  $x_2 = -1$

Output:  $\text{DELTA} = 0$ ; thiếu  $x_1$  và  $x_2$

## II. TEST-CASES

a.

a) How many test-cases we need for the following function  $f_1$ . What are they?

```
int f1(int x) {  
    if (x > 10)  
        return 2 * x;  
    else  
        return -x;  
}
```

Cần 2 test cases:

- Case 1:  $x > 10$
- Case 2:  $x \leq 10$

b.

b) Check if your test-cases can detect error if  $f_1$  is implemented as follows

```
int f1(int x) {  
    if (x > 10)  
        return 2 * x;  
    else if (x > 0)  
        return -x;  
    else  
        return 2 * x;  
}
```

In this case, how many test-cases we need to test this function? What are they?

Trong bài a. có 2 test cases:

- Case 1:  $x > 10$

Input:  $x = 11$

Expect: 22

Output: 22

→ ĐÚNG

- Case 2:  $x \leq 10$

Input:  $x = 10$

Expect: -10

Output: -10

→ ĐÚNG

→ Bộ test cũ không đủ để phát hiện ra hàm  $f_1$  này có lỗi vì thiếu branch coverage ở nhánh  $x \leq 0$

Để test bài b. cần 3 test cases:

- Case 1:  $x > 10$
- Case 2:  $0 < x \leq 10$

- Case 3:  $x \leq 0$

c.

c) How many test-cases we need to test this function? What are they?

```
int f2(int x) {
    if (x < 10)
        return 2 * x;
    else if (x < 2)
        return -x;
    else
        return 2 * x;
}
```

In this case, how many test-cases we need to test this function? What are they?

Vì branch  $(x < 2)$  sẽ luôn không được thực thi do branch  $(x < 10)$  nếu true đã kết thúc hàm  
 Vậy nên chương trình chỉ có 2 branches hoạt động:  $(x < 10)$  và  $(x \geq 10)$

Cần 2 test cases:

- Case 1:  $x < 10$
- Case 2:  $x \geq 10$

d.

d) How many test-cases we need to test this function? What are they?

```
int f3(int x) {
    if (log(x * x * cos(x)) < 3 * x)
        return 2 * x;
    else
        return 2 * x;
}
```

Vì cho dù điều kiện của branch  $(\log(x * x * \cos(x)) < 3 * x)$  là true hay false, hàm f3 sẽ luôn trả về  $2 * x \rightarrow$  Output không phụ thuộc vào điều kiện

Cần 1 test case:

- Case 1: any x

e.

e) Check if your test-cases can detect error if *findMax* is implemented as follows

```
int findMax(int num1, int num2, int num3) {
    int max = 0;
    if ((num1 > num2) && (num1 > num3))
        max = num1;
    if ((num2 > num1) && (num2 > num3))
        max = num2;
    if ((num3 > num1) && (num3 > num2))
        max = num3;
    return max;
}
```

In this case, how many test-cases we need to test this function? What are they?

Cần 2 test cases:

- Case 1: any num

Input: num1 = 5; num2 = 2; num3 = 1

Expect: 5

Output: 5

$\rightarrow$  ĐÚNG

- Case 2: have at least two equals num

Input: num1 = 5; num2 = 5; num3 = 1

Expect: 5  
Output: 0  
→ SAI

### III. PRACTICE I:

#### 1. Mô tả bài toán:

Input: 3 số thực a, b, c.

Output:

- Nếu  $a = b = c = 0$ : chương trình in "Infinite solutions."
- Nếu phương trình không có nghiệm thực: in "No solution."
- Nếu có nghiệm thực: in số lượng nghiệm thực và danh sách nghiệm.

#### 2. Xây dựng kịch bản test:

Để đảm bảo branch coverage cho đoạn mã trên, cần xét đến true/false của từng nhánh:

**Nhánh 1:** ( $a==0, b==0, c==0$ )

```
if (a == 0 && b == 0 && c == 0) {  
    return -1;  
}
```

**Nhánh 2:** ( $a==0, b==0, c\neq 0$ )

```
if (a == 0 && b == 0) {  
    return 0;  
}
```

**Nhánh 3:** ( $a==0, b\neq 0, y<0$ )

```
if (y < 0) return 0;
```

**Nhánh 4:** ( $a==0, b\neq 0, y\geq 0$ )

```
x[0] = sqrt(y);  
x[1] = -sqrt(y);  
return 2;
```

**Nhánh 5:** ( $a\neq 0, \Delta < 0$ )

```
if (delta < 0) return 0;
```

**Nhánh 6:** ( $a\neq 0, \Delta \geq 0, y_1 \geq 0$ )

```
if (y1 >= 0)
```

**Nhánh 7:** ( $a\neq 0, \Delta \geq 0, y_2 \geq 0 \ \&\& \ y_2 \neq y_1$ )

```
if (y2 >= 0 && y2 != y1)
```

Case	Input (a, b, c)	Expect Output
1	(0, 0, 0)	Infinite solutions.
2	(0, 0, 1)	No solution.
3	(0, 1, 1)	No solution.
4	(0, 1, -1)	The equation has 2 real solution(s): 1 -1
5	(1, 0, 1)	No solution.
6	(1, 0, -1)	The equation has 2 real solution(s): 1 -1
7	(1, -2, 1)	The equation has 2 real solution(s): 1 -1