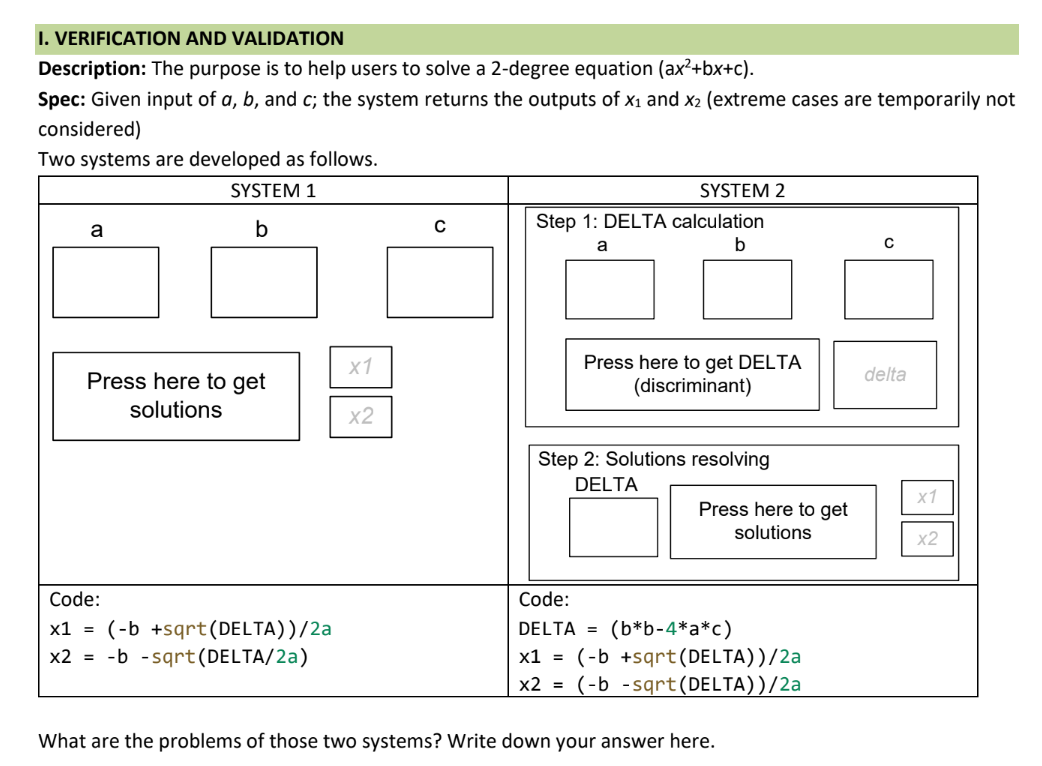
EXERCISE 1

NAME: Nguyễn Thành Đạt

STUDENT CODE : 3122411039

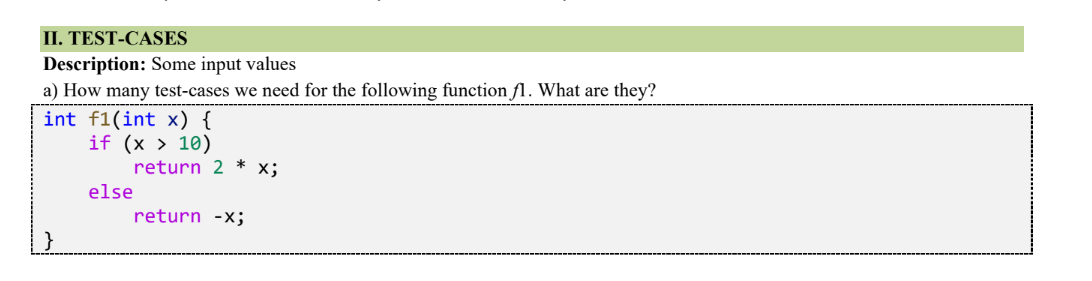


System 1:

The implemented formula does not strictly follow the correct mathematical expression. As a result, discrepancies appear in the discriminant (delta) because the system does not explicitly handle it. In addition, the formulas for x1 and x2 differ in terms of parentheses placement, which leads to incorrect results.

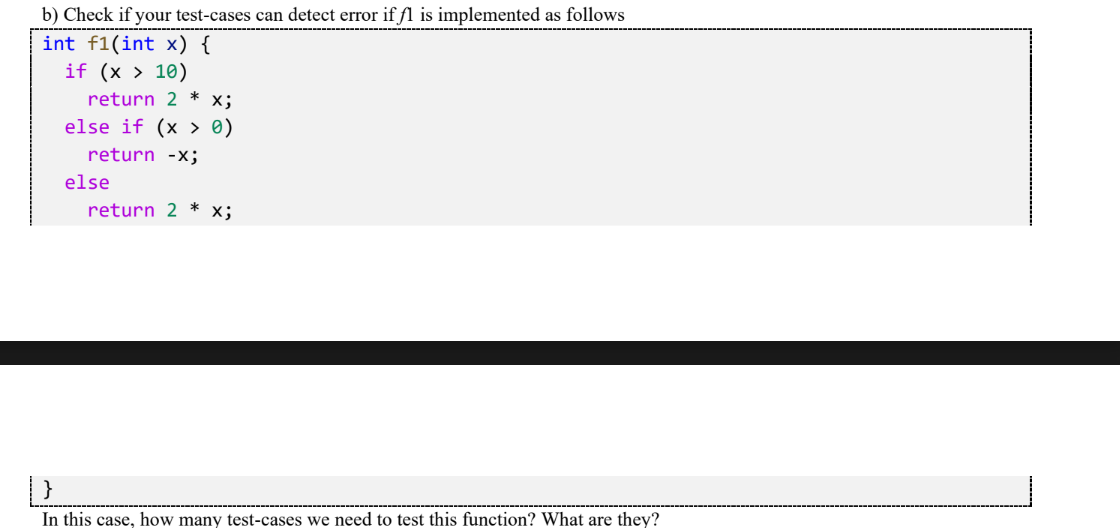
System 2:

The implementation does not cover all necessary cases. Specifically, it lacks handling for scenarios where delta < 0 (no real solutions) and where a = 0 (the equation is no longer quadratic).



| **TC** | **Input (x)** | **Expected Output** | **Nhánh được kiểm thử** |
| --- | --- | --- | --- |
| 1 | 11 | 22 | x > 10 |
| 2 | 8 | -8 | x ≤ 10 |

TEST CASE : 2



Check error with f1 ở trên

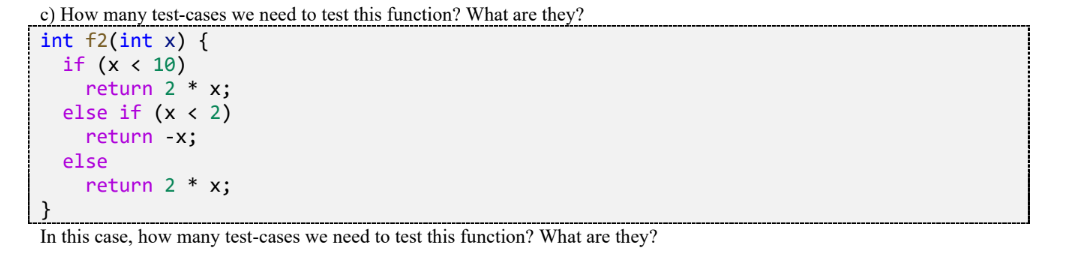
**f1 gốc:**

* Nếu x>10 → trả về 2x
* Nếu x≤10 → trả về −x

**f1 bị sửa:**

* Nếu x>10 → trả về 2x (đúng)
* Nếu 0<x≤10→ trả về −x (giống bản gốc)
* Nếu x≤0→ trả về 2x (sai, vì bản gốc trả về −x)

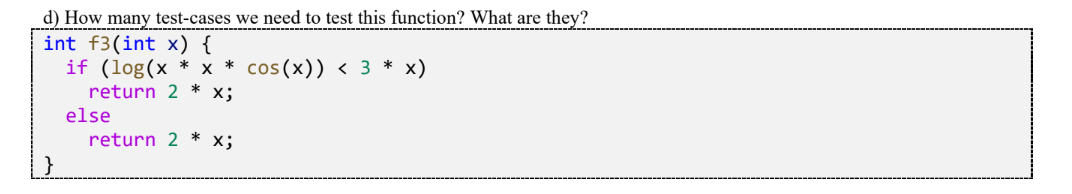
| **TC** | **Input (x)** | **Expected Output** | **Actual Output** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | 11 | 22 | 22 | Pass |
| 2 | 5 | -5 | -5 | Pass |
| 3 | 0 | 0 | 0 | Pass |
| 4 | -3 | 3 | -6 | **Fail** |



Lỗi logic: điều kiện x < 2 bị đặt **sau** x < 10 → nên nhánh đó **không bao giờ chạy**.

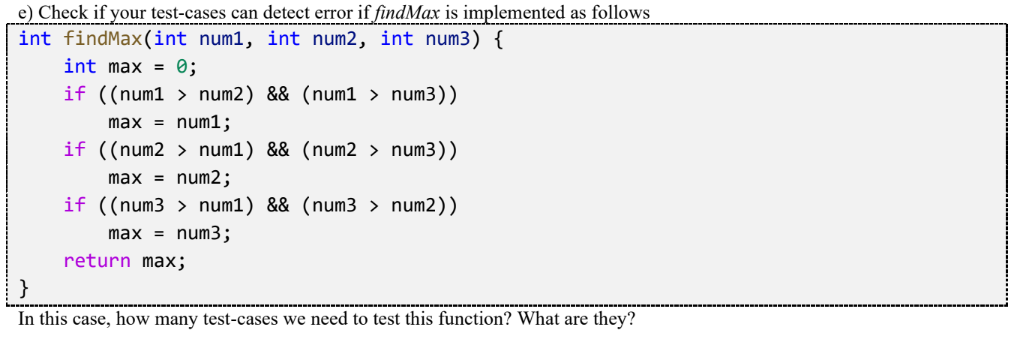
| **TC** | **Input (x)** | **Expected Output** | **Actual Output** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | 1 | -1 | 2 | **Fail** |
| 2 | 5 | 10 | 10 | Pass |
| 3 | 15 | 30 | 30 | Pass |

TEST CASE : 3



| **TC** | **Input (x)** | **Expected Output** | **Actual Output** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 4 | Pass |
| 2 | 0 | 0 | 0 | Pass |
| 3 | -1 | -2 | -2 | Pass |

TEST CASE :3



| **TC** | **Input (num1, num2, num3)** | **Expected Output** | **Actual Output** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | (5, 2, 1) | 5 | 5 | Pass |
| 2 | (2, 9, 3) | 9 | 9 | Pass |
| 3 | (1, 4, 7) | 7 | 7 | Pass |
| 4 | (5, 5, 3) | 5 | 0 | **Fail** |
| 5 | (2, 1, 1) | 2 | 2 | Pass |
| 6 | (5, 3,5 ) | 5 | 0 | **Fail** |
| 7 | (3,5,5) | 5 | 0 | **Fail** |
| 8 | (-7,-7,-7) | -7 | 0 | **Fail** |
| 9 | (7, 7, 7) | 7 | 0 | **Fail** |

TEST CASE : 9

III. PRATICE 1

• Mô tả bài toán, các input / output có thể có của bài toán

• Xây dựng các test cases kiểm tra tính đúng đắn chương trình

• Viết đoạn mã tự động kiểm tra chương trình cho bên dưới đúng hay sai?

#include <iostream>

#include <cmath>

using namespace std;

int solveQuartic(double a, double b, double c, double x[])

{

    if (a == 0 && b == 0 && c == 0)

    {

        return -1;

    }

    if (a == 0 && b == 0)

    {

        return 0;

    }

    if (a == 0)

    {

        double y = -c / b;

        if (y < 0)

            return 0;

        x[0] = sqrt(y);

        x[1] = -sqrt(y);

        return 2;

    }

    double delta = b \* b - 4 \* a \* c;

    if (delta < 0)

        return 0;

    double y1 = (-b + sqrt(delta)) / (2 \* a);

    double y2 = (-b - sqrt(delta)) / (2 \* a);

    int count = 0;

    if (y1 >= 0)

    {

        x[count++] = sqrt(y1);

        x[count++] = -sqrt(y1);

    }

    if (y2 >= 0 && y2 != y1)

    {

        x[count++] = sqrt(y2);

        x[count++] = -sqrt(y2);

    }

    return count;

}

int main()

{

    double a, b, c;

    cin >> a >> b >> c;

    double x[4];

    int n = solveQuartic(a, b, c, x);

    if (n == -1)

    {

        cout << " Infinite solutions." << endl;

    }

    else if (n == 0)

    {

        cout << "No solution." << endl;

    }

    else

    {

        cout << " The equation has " << n << " real solution(s): ";

        for (int i = 0; i < n; i++)

        {

            cout << x[i] << " ";

        }

        cout << endl;

    }

    return 0;

}

**Input**: 3 số thực a,b,ca, b, ca,b,c (hệ số của phương trình).

Output:

"Infinite solutions." nếu mọi số thực xxx đều là nghiệm (trường hợp a=b=c=0).

"No solution." nếu không tồn tại nghiệm thực.

"The equation has n real solution(s): ..." với nnn là số nghiệm thực tìm được.

| **TC** | **Input (a, b, c)** | **Expected Output** | **Actual Output** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | (0, 0, 0) | Infinite solutions | Infinite solutions | Pass |
| 2 | (0, 0, 1) | No solution | No solution | Pass |
| 3 | (0, 1, 1) | No solution | No solution | Pass |
| 4 | (0 1 0) | 2 solutions: -0,0 | 2 solutions: -0,0 | Pass |
| 5 | (0, 1, -10) | 2 solutions: 3.16228 , -3.16228 | 2 solutions: 3.16228 , -3.16228 | Pass |
| 6 | (1, 2, 2) | No solution | No solution | Pass |
| 7 | (1, -2, 1) | 2 solutions: 1, -1 | 2 solutions: 1, -1 | Pass |
| 8 | (1, -5, 6) | 4 solutions: 1.73205 , -1.73205 , 1.41421 , -1.41421 | 4 solutions: 1.73205 , -1.73205 , 1.41421 , -1.41421 | Pass |
| 9 | (1, -2, 1) | 2 solutions: 1, -1 | 2 solutions: 1, -1 | Pass |
| 10 | (1, 0, 0) | 2 solutions: 0, -0 | 2 solutions: 0, -0 | Pass |
| 11 | (1,-4, 4) | 2 solutions: 1.41421, -1.41421 | 2 solutions: 1.41421, -1.41421 | Pass |

Đoạn mã tự động kiểm tra chương trình cho bên dưới đúng hay sai

#include <iostream>

#include <sstream>

#include <vector>

#include <tuple>

#include <cmath>

#include <string>

using namespace std;

// === Hàm giải phương trình gốc ===

int solveQuartic(double a, double b, double c, double x[])

{

    if (a == 0 && b == 0 && c == 0)

    {

        return -1; // vô số nghiệm

    }

    if (a == 0 && b == 0)

    {

        return 0; // vô nghiệm

    }

    if (a == 0)

    {

        double y = -c / b;

        if (y < 0)

            return 0;

        x[0] = sqrt(y);

        x[1] = -sqrt(y);

        return 2;

    }

    double delta = b \* b - 4 \* a \* c;

    if (delta < 0)

        return 0;

    double y1 = (-b + sqrt(delta)) / (2 \* a);

    double y2 = (-b - sqrt(delta)) / (2 \* a);

    int count = 0;

    if (y1 >= 0)

    {

        x[count++] = sqrt(y1);

        x[count++] = -sqrt(y1);

    }

    if (y2 >= 0 && y2 != y1)

    {

        x[count++] = sqrt(y2);

        x[count++] = -sqrt(y2);

    }

    return count;

}

// === Hàm chạy test case ===

bool runTest(double a, double b, double c, const string &expected)

{

    double x[4];

    int n = solveQuartic(a, b, c, x);

    stringstream ss;

    if (n == -1)

    {

        ss << "Infinite solutions.";

    }

    else if (n == 0)

    {

        ss << "No solution.";

    }

    else

    {

        ss << "The equation has " << n << " real solution(s): ";

        for (int i = 0; i < n; i++)

        {

            ss << round(x[i] \* 100000) / 100000 << " "; // làm tròn 5 chữ số

        }

    }

    string result = ss.str();

    cout << "Input: (" << a << ", " << b << ", " << c << ")\n";

    cout << "Expected: " << expected << "\n";

    cout << "Actual:   " << result << "\n";

    if (result.find(expected) != string::npos)

    {

        cout << "Result: ✅ PASS\n\n";

        return true;

    }

    else

    {

        cout << "Result: ❌ FAIL\n\n";

        return false;

    }

}

// === Main test ===

int main()

{

    vector<tuple<double, double, double, string>> tests = {

    {0, 0, 0, "Infinite solutions."},

    {0, 0, 1, "No solution."},

    {0, 1, 1, "No solution."},

    {0, 1, 0, "The equation has 2 real solution(s):"},

    {0, 1, -10, "The equation has 2 real solution(s):"},

    {1, 2, 2, "No solution."},

    {1, -2, 1, "The equation has 2 real solution(s):"},

    {1, -5, 6, "The equation has 4 real solution(s):"},

    {1, -2, 1, "The equation has 2 real solution(s):"},

    {1, 0, 0, "The equation has 2 real solution(s):"},

    {1, -4, 4, "The equation has 2 real solution(s):"}

    };

    int passed = 0;

    for (int i = 0; i < tests.size(); i++)

    {

        auto [a, b, c, expected] = tests[i];

        if (runTest(a, b, c, expected))

        {

            passed++;

        }

    }

    cout << "Summary: " << passed << "/" << tests.size() << " test cases passed.\n";

    return 0;

}