**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA : CÔNG NGHỆ THÔNG TIN**

****

**Môn : Kiểm thử phần mềm**

**BÁO CÁO BÀI TẬP**

**Trà Đức Toàn - 3122411217**

**Lớp : DCT122C3**

**Thành phố Hồ Chí Minh , tháng 3 năm 2025**

Bài tập 1 :

Tên : Trà Đức Toàn

MSSV : 3122411217

A screenshot of a computer

AI-generated content may be incorrect.

Hệ thống 1 :

Tính x2 bị sai vì :

(DELTA/2a) là chỉ có delta /2a thay vì cả cụm phải chia cho 2a

Ghi đúng sẽ là x2 = (-b - sqrt(DELTA)) / 2a và lỗi mẫu phải có ngoặc không thì hệ thống sẽ hiểu thành /2 và nhân a riêng biệt thay vì 2\*a

Hệ thống 2 :

và lỗi mẫu phải có ngoặc không thì hệ thống sẽ hiểu thành /2 và nhân a riêng biệt thay vì 2\*a

A white rectangular object with black text

AI-generated content may be incorrect.

Cần 3 test case để kiểm thử hàm :

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input | Branch | Output |
| TC1 | 11 | x > 10 | 22 |
| TC2 | 10 | x <= 10 ( biên ) | -10 |
| TC3 | -5 | x <= 10 ( âm ) | 5 |

A white rectangular object with black text

AI-generated content may be incorrect.

Với 3 test case thì

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input | Branch | Output |
| TC1 | 11 | x > 10 | 22 |
| TC2 | 10 | 0 < x <= 10 | -10 |
| TC3 | -5 | x <0 | -10 |

Xảy ra vấn đề ở test case 3 vì khi input -10 không cho ta số dương như ở hàm đầu tiên

A white rectangular object with black text

AI-generated content may be incorrect.

Có 2 test case vì chỉ có 2 branch được thực hiện khi x<10 và x>=10 ( vì x <2 là vô lí khi mà x<10 đã bao trọn điều kiện )

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input | Branch | Output |
| TC1 | 9 | x < 10 | 18 |
| TC2 | 10 | x >= 10 | 20 |

A white rectangular object with black text

AI-generated content may be incorrect.

Dù x là gì thì ta đều nhận được 2\*x nhưng với log thì dk log > 0 nên sinh 2 ra test case

TC1 : x \* x \* cos(x) > 0

Ví dụ: x = 1 (vì 1\*1\*cos(1) > 0).

F3(1) = 2

x \* x \* cos(x) <= 0.

Ví dụ: x = 2 (vì 4\*cos(2) < 0).

Kiểm thử xem chương trình có lỗi toán học hay không

A screenshot of a computer

AI-generated content may be incorrect.

Số lượng test case cần thiết cho hàm này là 5 vì cần kiểm tra các trường hợp

Mỗi thứ tự đều có số lớn nhất , tất cả số bằng nhau , hai số bằng nhau

TC1: (5, 3, 2) → kỳ vọng 5.

TC2: (3, 7, 2) → kỳ vọng 7.

TC3: (2, 3, 8) → kỳ vọng 8.

TC4: (5, 5, 5) → kỳ vọng 5, hàm trả về 0

TC5: (5, 5, 3) → kỳ vọng 5, hàm trả về 0

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.A white rectangular object with black text

AI-generated content may be incorrect.

Ta cần giải phương trình **trùng phương dạng tổng quát**:

a⋅x4+b⋅x2+c=0

Trong đó:

**Input:** 3 hệ số thực a, b, c.

**Output:** số nghiệm thực và giá trị nghiệm thực (nếu có).

Nếu phương trình có vô số nghiệm → trả về -1.

Nếu không có nghiệm thực → trả về 0.

Nếu có nghiệm thực → in ra số lượng và các nghiệm.

Ý tưởng:

Đặt y=x2y = x^2y=x2, ta có phương trình bậc hai theo yyy:

a⋅y2+b⋅y+c=

Giải phương trình bậc hai này để tìm y.

**2. Xây dựng test case**

Ta cần bao phủ các tình huống sau:

1. **Tất cả hệ số bằng 0:**  
   Input: 0 0 0 → Vô số nghiệm.
2. **a = b = 0, nhưng c ≠ 0:**  
   Input: 0 0 5 → Phương trình vô nghiệm.
3. **a = 0, phương trình bậc nhất theo y:**
   * Input: 0 2 -8 → Giải 2y−8=0 ⇒ y=4 ⇒ nghiệm x=±2
4. **Delta < 0, phương trình bậc hai vô nghiệm theo y:**  
   Input: 1 0 1 → y2+1=0⇒ vô nghiệm.
5. **Có 2 nghiệm y ≥ 0, sinh ra 4 nghiệm x:**  
   Input: 1 -5 4 → y2−5y+4=⇒ y1=4, y2=1 ⇒ nghiệm x=±2,±1
6. **Có nghiệm y âm bị loại:**  
   Input: 1 0 -1 → y2−1=0⇒ y1=1 (giữ lại), y2=-1 (loại bỏ) ⇒ nghiệm x=±
7. **Nghiệm kép y1=y2 ≥ 0:**  
   Input: 1 -2 1 → y2−2y+1=0⇒ y=1 (nghiệm kép) ⇒ nghiệm x=±1

**Mã code tự động :**

#include <iostream>

#include <vector>

#include <cmath>

#include <cassert>

using namespace std;

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) { // phương trình b\*y + c = 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;

}

bool checkEqual(double a, double b, double eps = 1e-6) {

return fabs(a - b) < eps;

}

void runTest(double a, double b, double c, vector<double> expected) {

double x[4];

int n = solveQuartic(a, b, c, x);

if (expected.empty()) {

// Không có nghiệm hoặc vô số nghiệm

assert(n == 0 || n == -1);

}

else {

assert(n == (int)expected.size());

for (double e : expected) {

bool found = false;

for (int i = 0; i < n; i++) {

if (checkEqual(x[i], e)) {

found = true; break;

}

}

assert(found); // nghiệm mong đợi phải xuất hiện

}

}

cout << "Test (" << a << "," << b << "," << c << ") passed.\n";

}

int main() {

runTest(0, 0, 0, {}); // vô số nghiệm

runTest(0, 0, 5, {}); // vô nghiệm

runTest(0, 2, -8, { 2, -2 }); // x = ±2

runTest(1, 0, 1, {}); // vô nghiệm

runTest(1, -5, 4, { 2, -2, 1, -1 }); // x = ±2, ±1

runTest(1, 0, -1, { 1, -1 }); // x = ±1

runTest(1, -2, 1, { 1, -1 }); // nghiệm kép y=1 ⇒ x=±1

cout << "All tests passed!\n";

return 0;

}