EXERCISE 1

Name: Âu Thái Khang StudentCode:3122411086

1. VERIFICATION AND VALIDATION

Vấn đề của System1:

* Về giao diện tuy có tuân theo yêu cầu của đặc tả của người dùng là giải phương trình bậc 2. Tuy nhiên hệ thống này là không khai báo delta là gì và tính toán ra sao ngoài ra còn sai về mặt công thức tính nghiệm của x1 và X2. Có thể nói rằng giao diện đúng với đặc tả nhưng kết quả lại sai với mong muốn của người dùng.

Vấn đề của System2:

* Mặc dù công thức tính nghiệm là chính xác nhưng giao diện lại không đúng với đặc tả. Đặc tả yêu cầu nhập a,b,c vào hệ thống,rồi hệ thống phải trả về x1 và x2 dựa trên input. Tuy nhiên, hệ thống khong trả về x1 và x2 ngay lập tức, mà trả về delta và bắt người dùng phải nhập delta thì mới trả về x1 và x2. Có thể nói rằng giao diện và cách xử lý không tuân theo đặc tả nhưng kết quả lại đúng với yêu cầu người dùng.

1. **TEST-CASES**

a/Có 2 testcase cần cho hàm f1, để có thể bao phủ được hết code:

* Testcase đầu tiên là giá trị x>10: x=12 với kết quả mong muốn là 24
* Test case thứ hai là giá trị x <=10: x=10 với kết quả mong muốn là -10

Các tập tin đính kèm:

* baia.cpp
* testbaia.ipynb

b/Dựa trên hàm f1 ở câu a, hàm f1 ở câu b được viết lại theo cách khác cần có 3 testcase để có thể bao phủ được hết code:

* Test case 1, x>10: x=12 với kết quả mong muốn là 24
* Test case 2, 0<x<=10: x=10 với kết quả mong muốn là -10
* Test case 3: x<=0: x=-1 với kết quả mong muốn là 1. Testcase này có output thật sự là -2=> f1 có lỗi do khác kết quả mong muốn của câu a nếu x = -1

Các tập tin đính kèm:

* baib.cpp
* testbaib.ipynb

c/

câu này chỉ có 2 testcase vì câu if đầu if x < 10 bao trùm cả else if x < 2 nên khối else if sẽ không bao giờ được chạy nên là chỉ có 2 khối if đầu và else cuối được chạy.

* Testcase 1 với giá trị x <10: x=-5, expected=-10
* Testcase 2 với giá trị x>=10: x=11, expected=22

Các tập tin đính kèm:

* Baic.cpp
* Testbaic.ipynb

d/

vì là hàm log nên bên trong của nó sẽ phải > 0 và vì chỉ có 1 if và 1 else nên chỉ cần có 2 testcase là đủ:

* Testcase 1, giá trị x sao cho : x=1, expected=2
* Testcase 2, giá trị x sao cho : x=-1, expected=-2

Các file đính kèm:

* baid.cpp
* testbaid.ipynb

e/

Ngoài việc các testcase phải cover được hết code (testcase chạy được vào các cầu điều kiện if), thì cũng cần phải test thử tổ hợp các if trong chương trình này. Gọi 1 là trạng thái câu If thỏa và 0 là ngược lại, có 3 câu if trong chương trình, nên số tổ hợp các if là 2\*2\*2=8. Xét từng tổ hợp và đưa ra testcase cho từng trường hợp:

* (1, 0, 0) (trong đó 0 ở vị trí thứ i biểu thị câu if thứ i không thỏa, còn nếu là 1 thì if thỏa): a=5, b=2, c=1, expected=5
* (1, 1, 1): không thể xảy ra vì cả 3 số num1, num2, num3 đều lớn hơn nhau
* (1, 1, 0): không thể xảy ra vì num1 không thể vừa lớn hơn num2 và num2 lại lớn hơn num3
* (1, 0, 1): không thể xảy ra vì num1 không thể vừa lớn hơn num3 và num3 lại lớn hơn num1
* (0, 1, 1) : không thể xảy ra vì num2 không thể vừa lớn hơn num3 và num3 lại lớn hơn num2
* (0, 1, 0) : a=-2, b=0, c=-1, expected=0
* (0, 0, 1) : a=2, b=1, c=5, expected=5
* (0,0,0) a=5, b=5, c=5, expected=5

Ta có thể thấy được là chỉ có 4 cái thỏa mãn :

* (1, 0, 0) (trong đó 0 ở vị trí thứ i biểu thị câu if thứ i không thỏa, còn nếu là 1 thì if thỏa): a=8, b=3, c=2, expected=8
* (0, 1, 0) : a=-2, b=2, c=-1, expected=2
* (0, 0, 1) : a=2, b=1, c=6, expected=6
* (0,0,0): a=5, b=5, c=5, expected=5(Lỗi xảy ra ở testcase này, kết quả thực sự ra 0 thay vì 5)

Các file đính kèm:

* Baie.cpp
* Testbaie.ipynb

1. **PRATICE 1**

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

* Dựa theo code của hàm solveQuartic, chương trình này đường dùng để tìm x của phương trình với .
* Các input có thể có là:
  + a=0, , c tự do=> bài toán có dạng
  + a=b=0, => bài toán có dạng c=0
  + a=b=c=0 => bài toán có dạng 0=0
  + , b và c tự do=> bài toán có dạng
* Các output có thể có là:
  + Vô nghiệm
  + Vô số nghiệm
  + Có 1 nghiệm
  + Có 2 nghiệm
  + Có 3 nghiệm
  + Có 4 nghiệm

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

* Testcase 1: a=b=c=0, expected: vô số nghiệm
* Testcase 2 cho a=b=0, c0: a=b=0, c=5 expected: vô nghiệm
* Testcase 3 cho a=0, , c tự do với nhỏ hơn 0: a=0, b=-5, c=-2, expected: vô nghiệm
* Testcase 4 cho a=0, b0, c tự do với y=-c/b lớn hơn hoặc bằng 0: a=c=0, b=-7, expected: có một nghiệm là 0. Lỗi xảy ra: output thật sự là “The equation has 2 real solution(s): -0 0”
* Testcase 5 cho a0, delta>0, <=0: a=7, b=8, c=1, expected: vô nghiệm
* Testcase 6 cho a0, delta>0,: a=-1, b=2, c=35, expected: có hai nghiệm là,
* Testcase 7 cho a, delta=0, y<0: a=16,b=32,c=16, expected: vô nghiệm
* Testcase 8 cho a, delta=0, y>0: a=16, b=-32, c=16, expected: có 2 nghiệm là -1,1
* Testcase 9 cho a, delta<0: a=7, b=2, c=4, expected: vô nghiệm

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

*!*g*++* solveQuartic.cpp -o solveQuartic

!./solveQuartic < t1.inp > t1.out

*!*echo "Expected: Infinite solutions"

*!*cat t1.out

!./solveQuartic < t2.inp > t2.out

*!*echo "Expected: No solution"

*!*cat t2.out

!./solveQuartic < t3.inp > t3.out

*!*echo "Expected: No solutions"

*!*cat t3.out

!./solveQuartic < t4.inp > t4.out

*!*echo "Expected: The equation has 1 real solution(s): 0"

*!*cat t4.out

!./solveQuartic < t5.inp > t5.out

*!*echo "Expected: No solutions"

*!*cat t5.out

!./solveQuartic < t6.inp > t6.out

*!*echo "Expected: The equation has 2 real solution(s): 2.64575, -2.64575"

*!*cat t6.out

!./solveQuartic < t7.inp > t7.out

*!*echo "Expected: No solutions"

*!*cat t7.out

!./solveQuartic < t8.inp > t8.out

*!*echo "Expected: The equation has 2 real solution(s): 1, -1"

*!*cat t8.out

!./solveQuartic < t9.inp > t9.out

*!*echo "Expected: No solutions"

*!*cat t9.out

Các file đính kèm:

* solveQuartic.cpp
* testsolveQuartic.ipynb