

Phân tích và thiết kế thuật toán

Giảng viên: Nguyễn Thanh Sơn



Thành viên:

Hoàng Xuân Vũ

Đỗ Nguyễn Hoàng Huy

Nguyễn Đạt Huy Hoàng

Nguyễn Văn Tài



CS112.L21

Chủ đề: Kiểm tra tính đúng đắn của chương trình bằng bộ test

Testcase là gì?

- Là quá trình thử nghiệm, kiểm tra dữ liệu với đầu vào và đầu ra được xác định cụ thể.
- Là công cụ giúp cho chương trình, phần mềm hoạt động hiệu quả, chính xác và tối ưu.
- Là tạo phẩm giúp cho những lập trình viên tạo ra một sản phẩm hoàn chỉnh

Hai cách thực hiện test:

- ***White box:*** người kiểm thử biết được mã nguồn, thuật toán để kiểm thử (các trường hợp có thể bị sót do cài đặt)
- ***Black box:*** Người kiểm tra không biết được bộ dữ liệu (kiểm thử độc lập)

Tại sao phải tạo testcase?

- Để chương trình, phần mềm thực hiện đúng các yêu cầu, thủ tục được đưa ra.
- Kiểm tra lỗi, độ chính xác
- Tối ưu chương trình, phần mềm,...
- Tạo ra một sản phẩm hoàn thiện
- Biết được hiệu năng chương trình

Một chương trình hiệu quả cần:

Giải thuật đúng
đắn

+

Tốn ít tài
nguyên sử
dụng



Để kiểm tra tính hiệu quả của chương trình, người ta sử dụng testcase

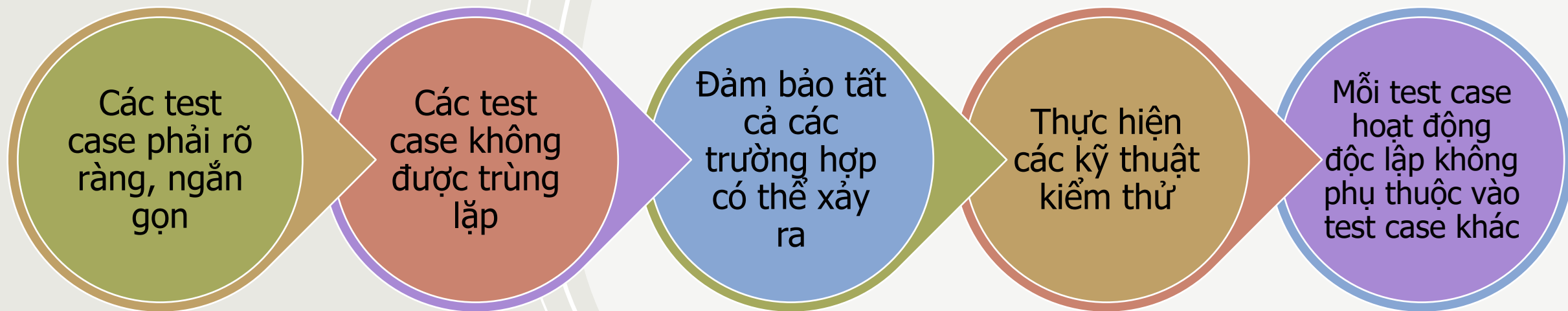
Thế nào là một testcase tốt?

- Dễ viết: Có thể bao quát nhiều trường hợp kiểm thử mà không mất nhiều công sức
- Dễ đọc: Mô tả chính xác được hành vi, chức năng kiểm thử
- Đồng nhất: Luôn trả về kết quả có cùng tính chất với yêu cầu đưa ra
- Khi kết quả kiểm thử thất bại, có thể dễ dàng tìm ra giá trị mong đợi và nhanh chóng xác định được vấn đề
- Đo được hiệu năng của chương trình

Cấu trúc cơ bản của một testcase:

- Test case ID: giá trị cần để xác định số lượng trường hợp để kiểm thử
- Test data: Dữ liệu cần chuẩn bị để test
- Test steps: Mô tả các bước thực hiện
- Expected results: Kết quả mong đợi từ các bước thực hiện trên (PASS nếu khớp với kết quả thực tế hoặc FAIL nếu khác)
- Thời gian chạy chương trình, bộ nhớ sử dụng

Các lưu ý cần thiết để viết một testcase hiệu quả





Ngoài ra testcase phải đảm bảo đủ các trường hợp:

- + Trường hợp phổ quát
- + Các điều kiện biên
- + Các trường hợp đặc biệt

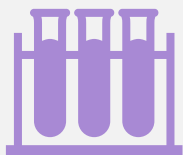
Một số lưu ý khi tạo test case



- Quy ước đặt tên cho test case: thường đặt theo chức năng hay trường hợp cụ thể để người đọc cũng như người sử dụng biết được



- Thứ tự thực hiện mỗi test case không ảnh hưởng đến kết quả thực thi.



- Chuẩn bị các dữ liệu test



Ước lượng và nơi rộng kích cỡ bộ dữ liệu để biết được ngưỡng dữ liệu của chương trình là bao nhiêu, mỗi cách giải sẽ cho mức ngưỡng để chương trình thực thi



- Nếu tạo test tự động thì khi viết chương trình sinh dữ liệu tự động phải đảm bảo input theo mô tả, output tương ứng và công thức đảm bảo output đúng

Áp dụng trong các bài toán

Hãy viết chương trình tìm các nghiệm của một phương trình: $ax^2+bx+c=0$ rồi in ra màn hình các nghiệm đó.

1. Input: Nhập vào trên cùng một dòng 3 số nguyên a , b , c tương ứng với các hệ số của phương trình.
2. Output: In ra kết quả nghiệm của phương trình.

Các trường hợp có thể xảy ra:

a=0:

$b = 0, c \neq 0$: Xuất ra màn hình "VN"

$b = 0, c = 0$: Xuất ra màn hình "VSN"

$b \neq 0$: Phương trình có 1 nghiệm $x = \frac{-c}{b}$

Các trường hợp có thể xảy ra:

+ $a \neq 0$:

$$\Delta = b^2 - 4ac$$

- $\Delta < 0$:

Phương trình vô nghiệm

- $\Delta = 0$:

Phương trình có 1 nghiệm duy nhất: $x = \frac{-b}{2a}$

- $\Delta > 0$:

Phương trình có 2 nghiệm phân biệt:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Bộ test mẫu:

Input	Output
0 0 0	VSN
0 0 1	VN
0 1 0	0
0 1 2	-2
1 0 0	0
1 0 1	VN
1 1 0	-1 0
1 -5 6	2 3
1 -8 -2	-0.24 8.24
10000 -68670 105041	2.3 4.57
1000 -5678 7356	2 3.68
1000000 -4251000 4312994	1.67 2.58
2 -4 2	1

Áp dụng trong các bài toán

Thay đổi hiển thị thời gian từ định dạng giờ 12 sang định dạng giờ 24

1. Input: Nhập vào thời gian theo định dạng giờ 12
2. Output: Xuất ra màn hình thời gian theo định dạng giờ 24

Input Format

Nhập vào một chuỗi là thời gian tính theo giờ 12 có AM và PM theo format h:m:sAM (hay h:m:sPM) ($h, m, s \in \mathbb{N}$)

Output Format

Xuất ra màn hình thời gian theo định dạng h:m:s **Lưu ý: Nếu định dạng đầu vào không đúng sẽ xuất ra Error

Bộ test mẫu:

Input	Output
00:00:00PM	Error
12:00:00AM	00:00:00
12:45:15PM	12:45:15
07:40:23PM	19:40:23
11:58:13AM	11:58:13
13:15:45AM	Error
13:15:45PM	Error

Áp dụng trong các bài toán

Thực hiện phép tính chia hai phân số

- Input: Nhập vào phân số thứ nhất, phân số thứ hai
- Output: In ra kết quả thực hiện phép tính đó

Input Format

Nhập vào số nguyên a, b, c, d tương ứng với tử, mẫu của phân số thứ nhất và phân số thứ hai. a, b, c, d nằm trên cùng một dòng ($-10^5 \leq a, b, c, d \leq 10^5$)

Output Format

- Kết quả theo định dạng a/b
- Nếu kết quả không tồn tại thì xuất 'Error'

Các trường hợp có thể xảy ra:

- + Một trong 3 đầu vào $b, c, d = 0$

Xuất ra màn hình Error

- + a, b, c, d thỏa mãn đề bài:

Rút gọn phân số:

- + Phân số ở dạng tối giản a/b

- + Nếu $a.b < 0$ thì phải đặt dấu " – " ở trên tử số

- + Kết quả ở dạng số nguyên thì chỉ xuất ra số nguyên

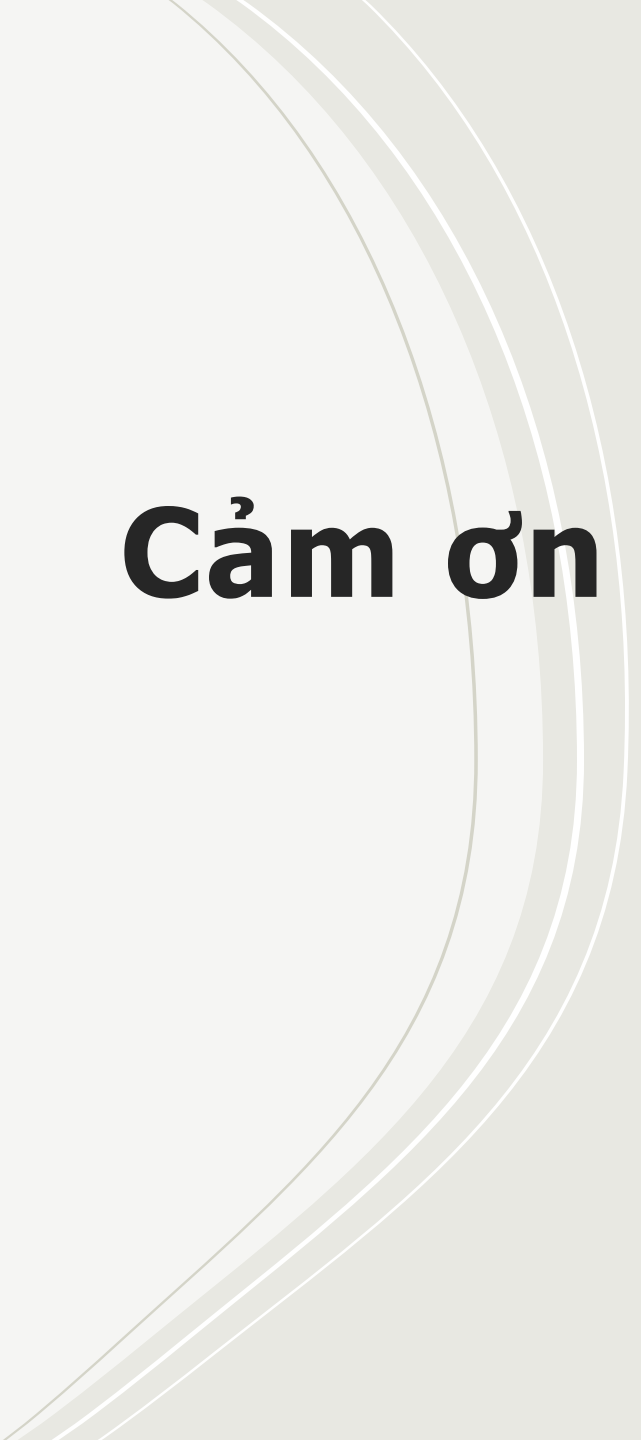
- + Kết quả ở dạng số vô tỉ thì làm tròn đến chữ số thập phân thứ hai

Bộ test mẫu:

Input	Output
1 1 0 9	Error
1 -2 3 6	-1
0 1 2 3	0
1 0 3 0	Error
3 4 5 -6	-9/10
-3 1 1 -3	9
-6 7 -6 7	1
2 -2 2 2	-1
-1 1 15 0	Error

Link bài tập mẫu và bài tập về nhà:

<https://www.hackerrank.com/contests/cs112-l21/challenges>



**Cảm ơn thầy và các bạn đã chú
ý lắng nghe**