Họ và tên: Chu Quang Tú

Mã sinh viên: 21020393

**Đề bài:** Nhập vào 2 số dương là chiều dài và chiều rộng của hình chữ nhật. Tính diện tích hình chữ nhật đó

Link github: https://github.com/ChuTu15/kiemthu.git

# **Xây dựng đồ thị điều khiển cho chương trình.**

A screenshot of a cell phone

Description automatically generatedĐơn vị chương trình là một hàm có tên là area nhận vào hai tham số là chiều dài và chiều rộng của hình chữ nhật và trả về diện tích của nó. Đồ thị điều khiển có thể được vẽ như sau:

# **Lựa chọn tiêu chí kiểm thử luồng dữ liệu**

1. Với các biến chiều dài (length), chiều rộng (width) và diện tích (S), ta có:

* length: def(length) = {B}; c-use(length) = {D}; p-use(length) = {C}
* width: def(width) = {B}; c-use(width) = {D}; p-use(width) = {C}
* S: def(S) = {D}; c-use(S) = {F}; p-use(S) = {}

1. Các du-pair:

* length: {B, D}
* width: {B, D}
* S: {D, F}

# **Xác định các đường đi trên CFG thoả mãn tiêu chí kiểm thử đã chọn**

Dựa vào các tiêu chí đã xác định ở trên:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Du-pair | Def-clear-path | Complete path |
| length | (B, D) | B, C(T), D | B, C(T), D, F, G |
| (B, D) | B, C(F), E | B, C(F), E, G |
| width | (B, D) | B, C(T), D | B, C(T), D, F, G |
| (B, D) | B, C(F), E | B, C(F), E, G |
| S | (D, F) | D, F | D, F, G |

# **Sinh các ca kiểm thử**

Có 6 testcases tương ứng với 2 complete path như sau:

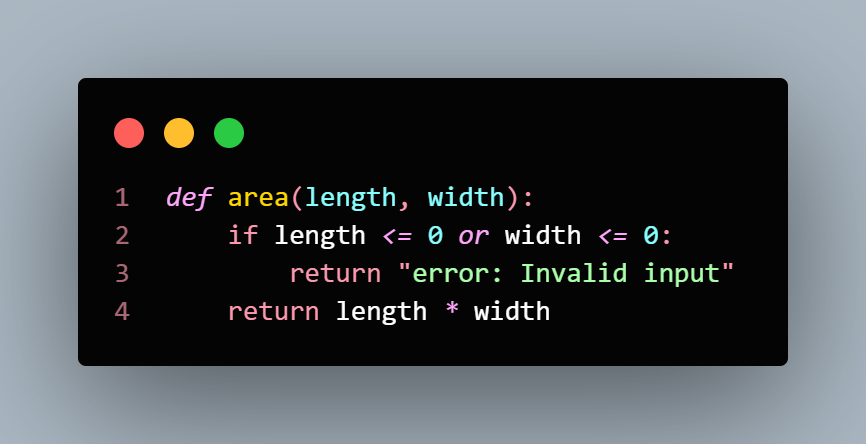
1. (B, C(T), D, F, G) với các ca kiểm thử trả về được giá trị của S:

* length và width đều là số dương: {length: 10, width: 10} => S = 100

1. (B, C(F), E, G) với các ca kiểm thử cho trường hợp báo lỗi:

* length và width đều là số âm: {length: -10, width: -10}
* length là số dương và width là số âm: {length: 10, width: -10}
* length là số âm và width là số dương: {length: -10, width: 10}
* length hoặc width bằng 0: {length: 0, width: 10}; {length: 10, width: 0}

# **Thực hiện code và kiểm thử**

1. Công nghệ sử dụng
   1. Ngôn ngữ lập trình: Python.
   2. Kiểm thử: sử dụng thư viện unittest của python.
2. Viết chương trình tính diện tích hình chữ nhật
3. Thực hiện các testcase bằng unittest
4. A black background with a black square

   Description automatically generated with medium confidenceKết quả

# **Kết luận**

Chương trình tính toán diện tích của hình chữ nhật đã đáp ứng được các yêu cầu**.**

# **Bài tập**

Bài 1:

Trong quy trình kiểm thử dòng dữ liệu động, gồm các bước như sau:

* Vẽ đồ thị luồng điều khiển (CFG)
* Lựa chọn tiêu chí kiểm thử luồng dữ liệu
* Xác định các đường đi trên CFG thoả mãn tiêu chí kiểm thử đã chọn
* Sinh các ca kiểm thử tương ứng

Bài 2:

1. Đồ thị luồng điều khiển:

A diagram of a program

Description automatically generated

1. def(X) = {1, 5}, p-use(X) = {3}, c-use(X) = {4, 6}

def(Y) = {1,4}, p-use(Y) = {2}, c-use(Y) = {4, 6}

Các du-pairs của biến X là:

1 – 3T, 1 – 3F, 1 – 4, 1 – 6, 5 – 3T, 5 – 3F, 5 – 4, 5 - 6

Các du-pairs của biến Y là:

1 – 2(Y>0), 1 – 2(Y<=0), 1 – 4, 1 – 6, 4 – 2(Y>0), 4 – 2(Y<=0), 4 – 6

Với biến X

|  |  |  |
| --- | --- | --- |
| Du-pairs | Đường đi | Testcase all-use |
| 1 – 3T | 1 – 2(Y>0) – 3T | X = 2, Y = 3 |
| 1 – 4 | 1 – 2(Y > 0) – 3T – 4 | X = 2, Y = 3 |
| 1 – 6 | 1 – 2(Y<=0) – 6 | X = 2, Y = -1 |
| 5 – 3T | 5 – 2(Y>0) – 3T | X = -1, Y = 2, X = 2 |
| 5 – 4 | 5 – 2(Y>0) – 3T – 4 | X = -1, Y = 2, X = 2 |
| 5 – 6 | 5 – 2(Y>0) – 3T – 4 – 2(Y<=0) - 6 | X = -1, Y = 2, X = 3 |

Với biến Y

|  |  |  |
| --- | --- | --- |
| Du-pairs | Đường đi | Testcase all-use |
| 1 – 2(Y>0) | 1 – 2(Y>0) | X = 2, Y = 3 |
| 1 – 4 | 1 – 2(Y>0) – 3T - 4 | X = 2, Y = 3 |
| 1 - 6 | 1 – 2(Y<=0) – 6 | X = 2, Y = -1 |
| 4 – 2(Y>0) | 4 – 2(Y>0) | X = 2, Y = 6 |
| 4 – 6 | 4 – 2(Y<=0) – 6 | X = 2, Y = 1 |

Bài 3:

1. Đồ thị dòng dữ liệu:

A diagram of a algorithm

Description automatically generated

Def(n) = {1}, p-use(n) = {4}, c-use(n) = {}

Def(result) = {2, 5}, p-use(result) = {}, c-use(result) = {5, 7}

Def(i) = {3, 6}, p-use(i) = {4}, c-use = {5, 6}

Bài 4:

Với biến x:

|  |  |  |  |
| --- | --- | --- | --- |
| Def-clear-path | Du-path | All-p-uses /some c-uses | All-c-uses / some p-uses |
| 0 – 1;  0 – 1 - 3 0 – 1 – 2;  0 – 1 – 2 – 4;  0 – 1 – 2 – 4 – 5;  0 – 1 – 2 – 4 – 6;  0 – 1 – 2 – 4 – 5 – 6; 3 – 4; 3 – 4 – 5; 3 – 4 – 5 – 6; 3 – 4 – 6; | x x  x x   x x | x   x    x | x   x    x |

Với biến y:

|  |  |  |  |
| --- | --- | --- | --- |
| Def-clear-path | Du-path | All-p-uses /some c-uses | All-c-uses / some p-uses |
| 0 – 1; 0 – 1 – 3; 0 – 1 – 3 – 4; 0 – 1 – 3 – 4 - 6; 2 – 4; 2 – 4 – 6; 5 – 6; | x  x x x x x | x  x  x | x  x x |

1. Đường đi (0 – 1 – 3 – 4 – 5 – 6) khả thi vì:

ban đầu nhập X = 2, Y = 2. Do đó x + y = 4 nên sẽ đi đến 3.

Từ 3, ta sẽ khởi tạo X = 3, và Y = X + 1. Tức là X = 3, Y = 3.

Ở 4, vì x^2 + y^2 = 9 + 9 = 18 nên sẽ đi đến 5.

Ở 5, ta đặt Y = X + 1. Sau đó đi đến 6.

1. Vì đó là 1 lệnh với use được sử dụng trước def.

Ví dụ: x = x + 1.

Bài 5:

A diagram of a algorithm

Description automatically generatedCFG:

Đường đi và các ca kiểm thử với độ đo C2:

Path1: 1 – 2(T) – 3 – 4(T) – 5 – 6(F) – 8(F) – 10 – 11(F) –13 - 10 - 14.  
Testcase: -3 -6

Path2: 1 – 2F – 4F – 6T – 7  
Testcase: 0 4

Path3: 1 – 2F – 4F – 6F – 8(T) – 9  
Testcase: 4 0

Path4: 1 – 2F – 4F – 6F – 8F – 10 – 11T – 12 – 10 – 14  
Testcase: 6 3

Bcov = 10/10 = 1

Đường đi và các ca kiểm thử với độ đo all-def coverage:

|  |  |  |
| --- | --- | --- |
| Biến | Đường đi | Test-case |
| m | 1 – 2(T) - 3 | m = -1, n = 2 |
| m | 3 – 4(F) – 6 | m = -1, n = 2 |
| m | 12 – 10 – 14 | m = 6, n = 3 |
| n | 1 – 2(F) – 4 | m = 1, n = 2 |
| n | 5 – 6(F) – 8 | m = 1, n = -3 |
| n | 13 – 10 – 14 | m = 3, n = 6 |