

Đỗ Đức Thắng – 23020158

Link github: <https://github.com/V1ctorious3010/KiemThuPhanMem>

Câu 1. Trình bày các bước trong kiểm thử dòng dữ liệu tự động

Quy trình:

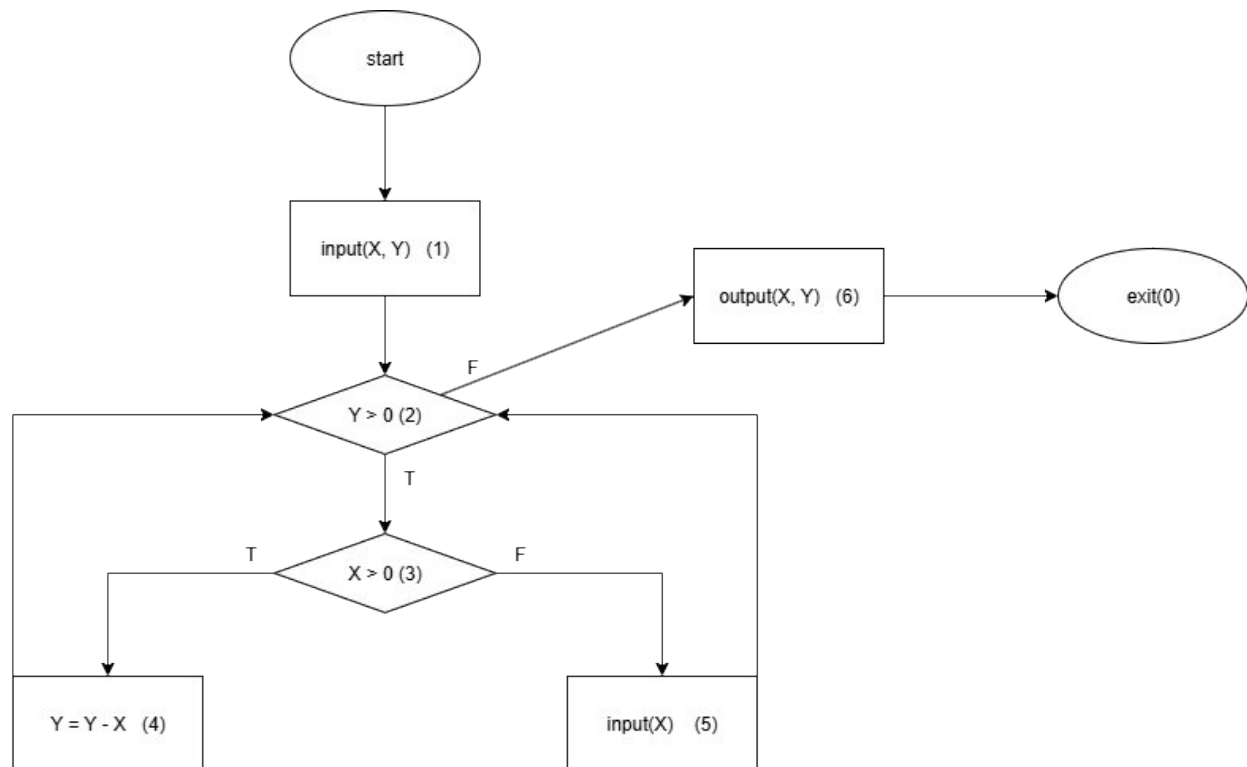
B1. Xây dựng luồng điều khiển CFG

B2. Lựa chọn độ đo kiểm thử

B3. Xác định đường đi trên CFG thỏa mãn tiêu chí kiểm thử đã chọn

B4. Sinh ra ca kiểm thử

Câu 2.



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

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

**Biến X**

STT	Du-pair	Đường def-clear	Đường đầy đủ	Test case(X,Y)
1	(1, 4)	1,2(T),3(T),4	1,2(T),3(T),4,2(F),7	(5,4)
2	(1, 6)	1,2(T),3(T),4,2(F),7	1,2(T),3(T),4,2(F),6	(2,7)

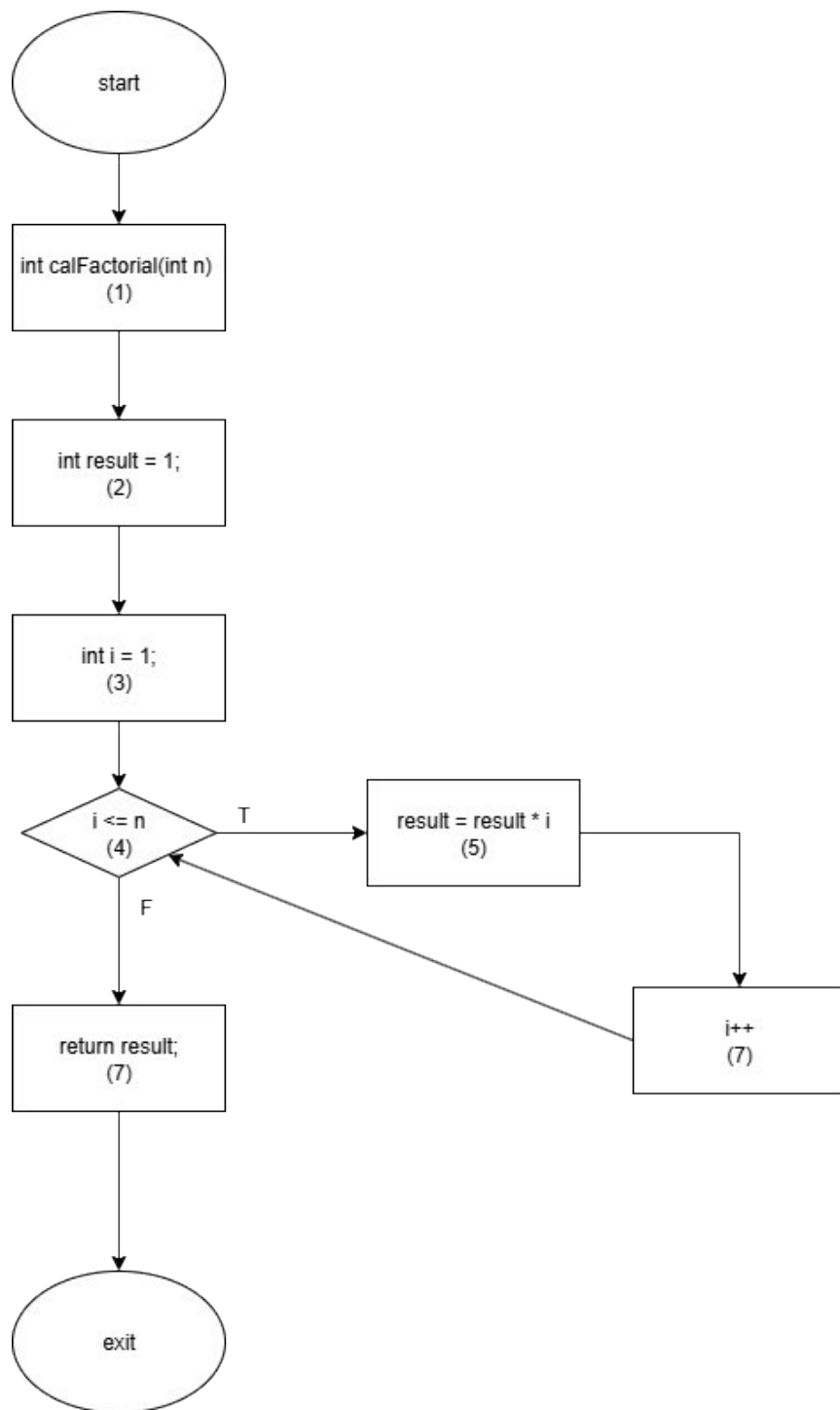
3	(1, 3(T))	1,2(T),3(T)	1,2(T),3(T),4,2(F),6	(4,9)
4	(1, 3(F))	1,2(T),3(F)	1,2(T),3(F),5,2(T),3(T),4,2,6	(0,5) (2)
5	(5, 4)	5,2(T),3(T),4	1,2(T),3(F),5,2(T),3(T),4,2,6s	(-2,6) (3)
6	(5, 7)	5,2(T),3(T),4,2(F),7	1,2(T),3(F),5,2(T),3(T),4,2,6	(-1,4) (5)
7	(5, 3(T))	5,2(T),3(T)	1,2(T),3(F),5,2(T),3(T),4,2,6	(-10,3) (2)
8	(5, 3(F))	5,2(T),3(F)	1,2(T),3(F),5,2(T),3(F),5,2,3(T),4,6	(-3,5) (0) (4)

### Biến Y

STT	Du-pair	Đường def-clear	Đường đầy đủ	Test case (X,Y)
1	(1, 4)	1,2(T),3(T),4	1,2(T),3(T),4,2(F),6	(6,5)
2	(1, 6)	1,2(T),3(T),4,2(F),6	1,2(T),3(T),4,2(F),6	(3,8)
3	(1, 2(T))	1,2(T)	1,2(T),3(T),4,2(T),6	(1,2)
4	(1, 2(F))	1,2(F)	1,2(F),6	(5,0)
5	(4, 4)	4,2(T),3(T),4	1,2(T),3(T),4,2(T),3(T),4,2(F),6	(2,9)
6	(4, 6)	4,2(F),6	1,2(T),3(T),4,2(F),6	(2,5)
7	(4, 2(T))	4,2(T)	1,2(T),3(T),4,2(T),3(T),4,2(F),6	(3,10)
8	(4, 2(F))	4,2(F)	1,2(T),3(T),4,2(F),6	(5,4)

C3.

result	def(n) = {2,5}	c-use={5,7}	p-use={}
n	def(n) = {1}	c-use={}	p-use={4}
i	def(i) = {3,6}	c-use={5,6}	p-use={4}



C4.

1,

- Def clear paths của biến X:

0,1

0,1,2

0,1,2,4

0,1,2,4,6

0,1,2,4,5

0,1,2,4,5,6

3,4

3,4,5

3,4,6

3,4,5,6

- Def clear path của biến Y:

0,1,3,4,6

2 + 3.

Biến	def	c-use	p-use
x	{0, 3}	{3, 5}	{1, 4}
y	{0, 2, 5}	{6}	{1, 4}

Biến X:

du-pair	du-path	Ghi chú
(0,1(T))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(0,1(F))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(0,3)	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all c-uses / some p-uses
(0,4(T))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(0,4(F))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all p-uses / some c-uses
(0,5)	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all c-uses / some p-uses
(3,4(T))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(3,4(F))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all p-uses / some c-uses
(3,5)	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all c-uses / some p-uses

Biến Y

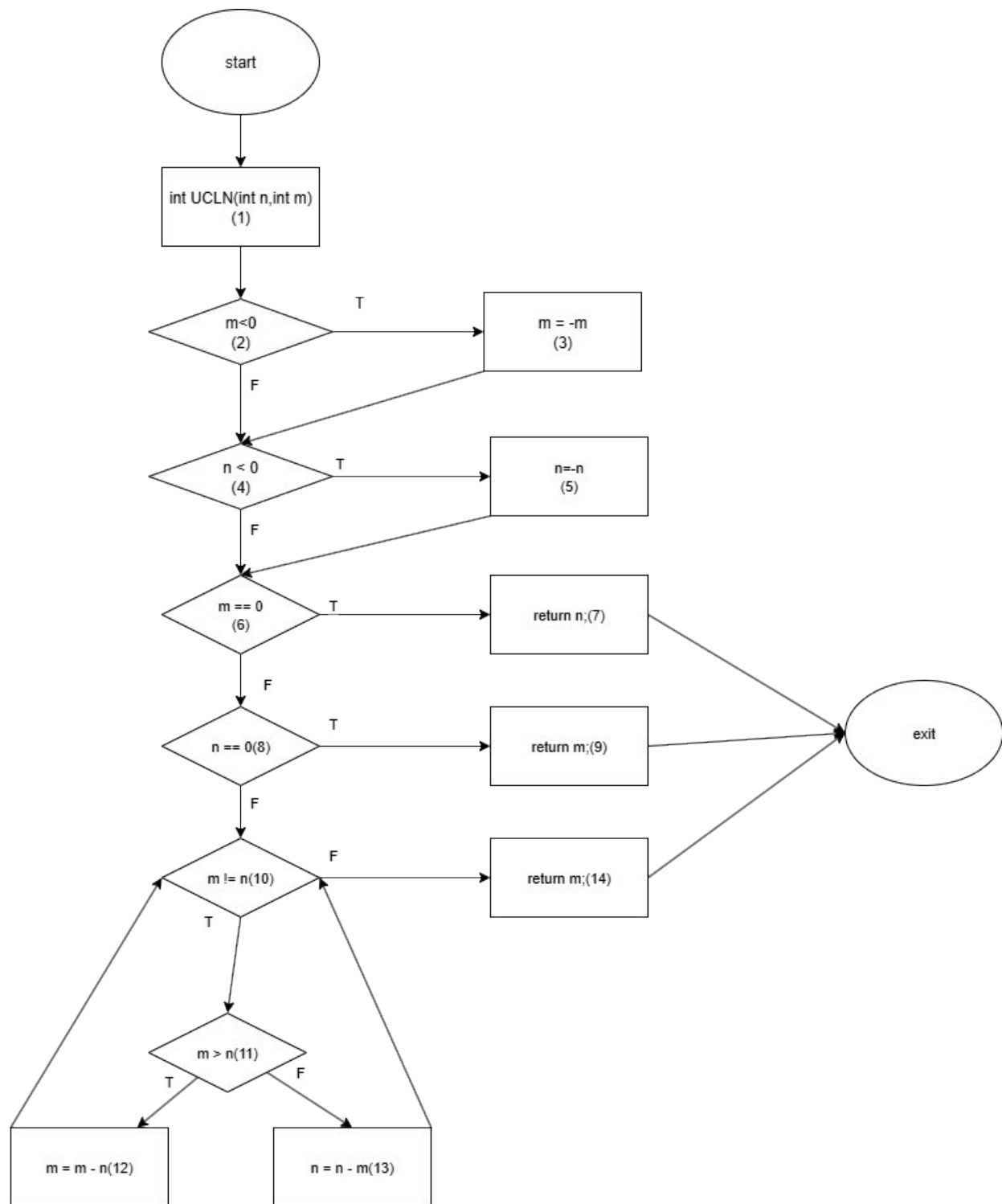
(0,1(T))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(0,1(F))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(0,4(T))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses

(0,4(F))	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all p-uses / some c-uses
(0,6)	$0 \rightarrow 1(F) \rightarrow 3 \rightarrow 4(T) \rightarrow 6$	all c-uses / some p-uses
(2,4(T))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(T) \rightarrow 6$	all p-uses / some c-uses
(2,4(F))	$0 \rightarrow 1(T) \rightarrow 2 \rightarrow 4(F) \rightarrow 5 \rightarrow 6$	all p-uses / some c-uses
(2,6)	$0 \rightarrow 1 \rightarrow 2 \rightarrow 4(T) \rightarrow 6$	all c-uses / some p-uses
(5,6)	$0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$	all c-uses / some p-uses

4. Với test case  $x = 6$  và  $y = -2$ , các điều kiện tại cạnh (1, 3) và (4, 5) đều được thỏa mãn, đường đi 0,1,3,4,5,6 được thực thi -> đường đi khả thi

5. Theo định nghĩa def-clear path, một def phải đi trước và không bị ghi đè trước khi tới use. Vì tại đỉnh 3 use diễn ra trước def nên def tại đỉnh này không thể đi tới use theo định nghĩa def-clear -> không tạo cặp def-use

C5.



Đường đi và ca kiểm thử với độ đo C2

Path

1, 2(F), 4(F), 6(T), 7

Test case: (0, 36)

Expected Output: 36

Path

1, 2(F), 4(F), 6(F), 8(T), 9

Test case: (36, 0)

Expected Output: 36

Path

1, 2(T), 3, 4(T), 5, 6(F), 8(F), 10(F), 14

Test case: (-36, -36)

Expected Output: 36

Path

1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(T), 12, 10(F), 14

Test case: (2, 1)

Expected Output: 1

Path

1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(F), 13, 10(F), 14

Test case: (1, 2)

Expected Output: 1

Path

1, 2(F), 4(F), 6(F), 8(F), 10(F), 14

Test case: (3, 3)

Expected Output: 2

**Sinh đường đi và các ca kiểm thử với độ đo All-def coverage**

**Biến m:**

- $\text{def}(m) = \{1, 3, 12\}$
- $\text{c-use}(m) = \{3, 9, 12, 13, 14\}$
- $\text{p-use}(m) = \{2, 6, 10, 11\}$

Các cặp du-pair và ca kiểm thử:

1. Du-pair (1, 3)

Def-clear path: 1, 2(T), 3

Complete path: 1, 2(T), 3, 4(F), 6(F), 8(T), 9

Test case: (-1, 0)

Expected Output: 1

2. Du-pair (3, 9)

Def-clear path: 3, 4(F), 6(F), 8(T), 9

Complete path: 1, 2(T), 3, 4(F), 6(F), 8(T), 9

Test case: (-1, 0)

Expected Output: 1

3. Du-pair (12, 14)

Def-clear path: 12, 10(F), 14

Complete path: 1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(T), 12, 10(F), 14

Test case: (36, 1)

Expected Output: 1

#### **Biến n:**

- $\text{def}(n) = \{1, 5, 13\}$
- $\text{c-use}(n) = \{5, 7, 13\}$
- $\text{p-use}(n) = \{4, 8, 10, 11\}$

Các cặp du-pair và ca kiểm thử:

1. Du-pair (1, 5)

Def-clear path: 1, 2(F), 4(T), 5

Complete path: 1, 2(F), 4(T), 5, 6(T), 7

Test case: (0, 36)

Expected Output: 36

2. Du-pair (5, 7)

Def-clear path: 5, 6(T), 7

Complete path: 1, 2(F), 4(T), 5, 6(T), 7

Test case: (0, 36)

Expected Output: 36

3. Du-pair (13, 10)

Def-clear path: 13, 10(F), 14

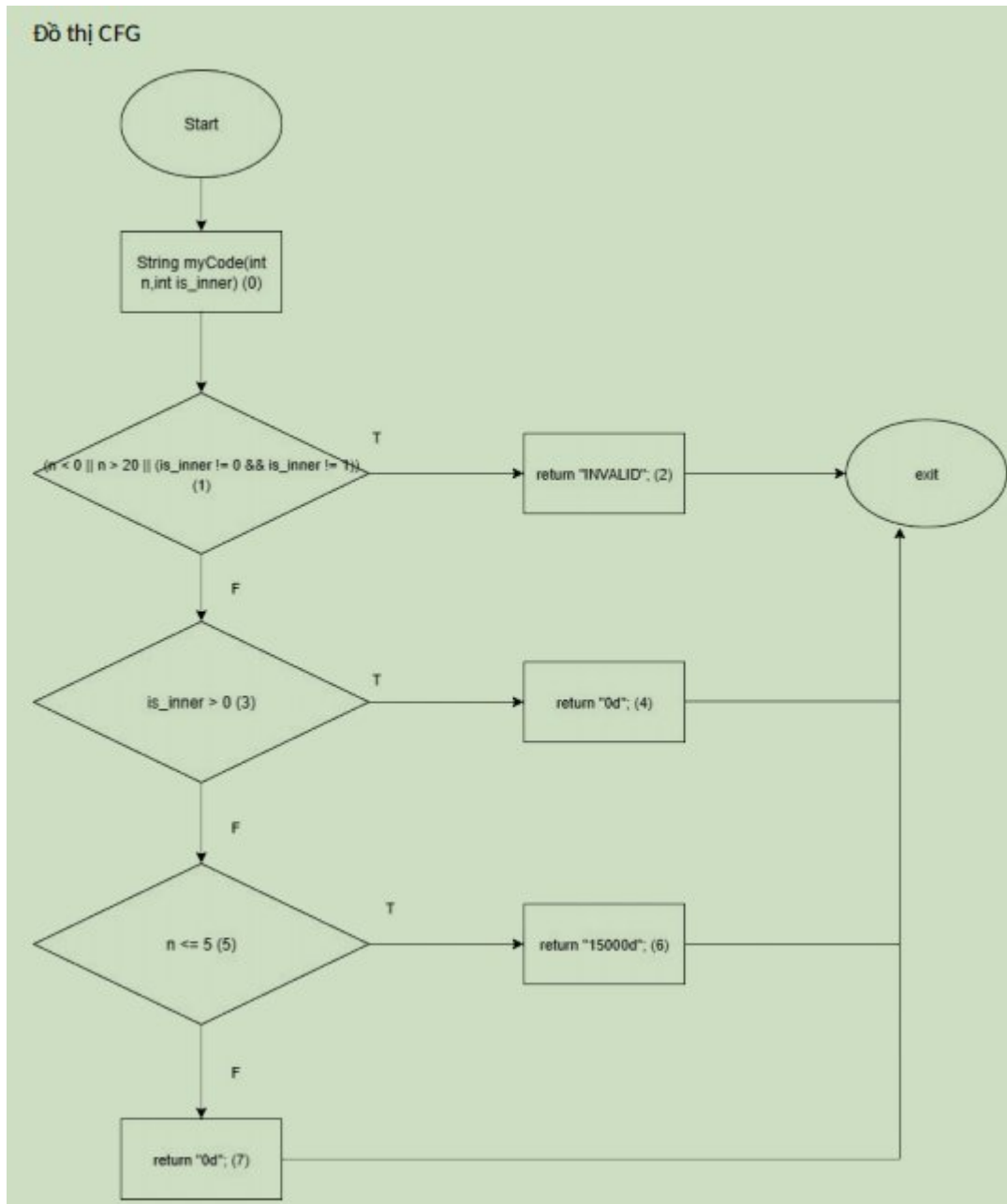


Complete path: 1, 2(F), 4(F), 6(F), 8(F), 10(T), 11(F), 13, 10(F), 14

Test case: (2, 4)

Expected Output: 2

C6(Bài làm cá nhân)



Thiết kế ca kiểm thử theo độ phủ all use

Variable	Du-pair	Def-clear path	Complete path	Test case	Expected Output	Actual Output	Result
n	(1,1)	1	1(T) → return (-1,0)	(-1,0)	"INVALID"	"INVALID"	Pass

n	(1,3T)	1→2(F)→3	1(F)→2(F)→3(T)→4	(4,0)	"15000d"	"15000d"	Pass
n	(1,3F)	1→2(F)→3	1(F)→2(F)→3(F)→5	(10,0)	"0d"	"0d"	Pass
is_inner	(1,1)	1	1(F)→2(F)→3(T)→4	(4,0)	"15000d"	"15000d"	Pass
is_inner	(1,2T)	1→2	1(F)→2(T)→return	(3,1)	"0d"	"0d"	Pass
is_inner	(1,2F)	1→2	1(F)→2(F)→3(F)→5	(10,0)	"0d"	"0d"	Pass

## Test

```
#include <bits/stdc++.h>
using namespace std;

string myCode(int n, int is_inner)
{
    if (n < 0 || n > 20 || (is_inner != 0 && is_inner != 1))
        return "INVALID";

    if (is_inner)
        return "0d";
    else if (n <= 5)
        return "15000d";
    else
        return "0d";
}

struct TestCase
{
    int n, is_inner;
    string expected;
};

int main()
{
    vector<TestCase> tests =
    {
        {-1, 0, "INVALID"},
        {4, 0, "15000d"},
        {10, 0, "0d"},
        {4, 0, "15000d"},
        {3, 1, "0d"},
        {10, 0, "0d"}
    };

    for (int i = 0; i < (int)tests.size(); i++)
    {
        string actual = myCode(tests[i].n, tests[i].is_inner);
        cout << "Test " << i + 1 << ": "
              << (actual == tests[i].expected ? "Pass" : "Fail") << "\n";
    }
}
```