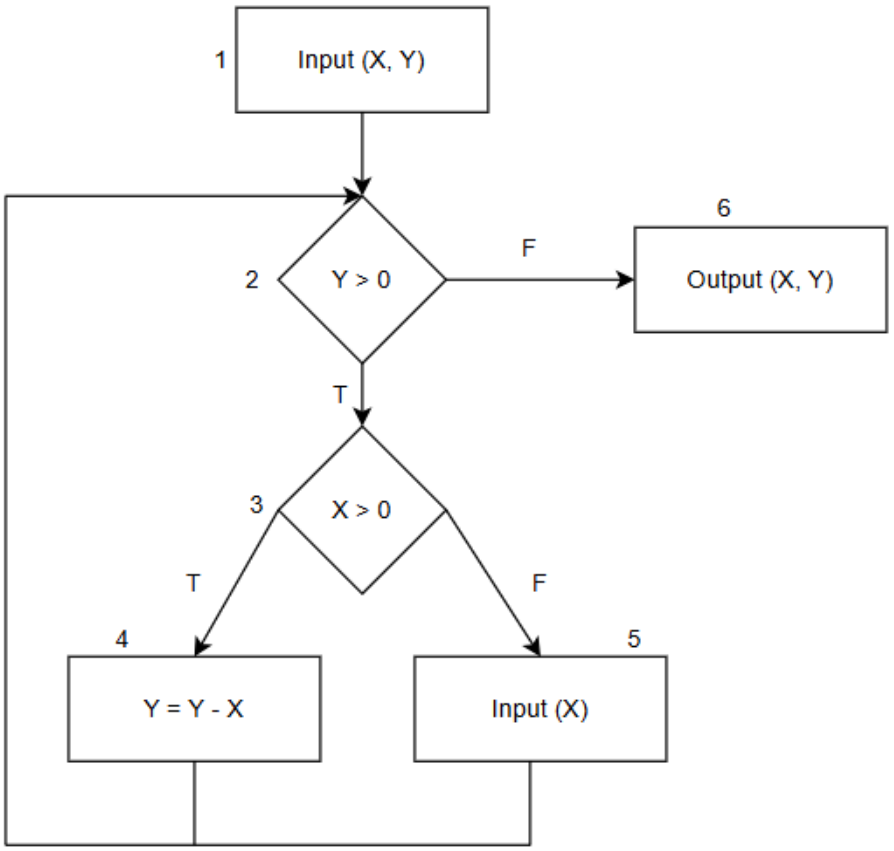
Bài 1

Các bước trong quy trình kiểm thử dòng dữ liệu động

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

Bài 2

1. *Control Flow Graph*



2. *Xác định du-pairs cho X và Y kết hợp ý 3*

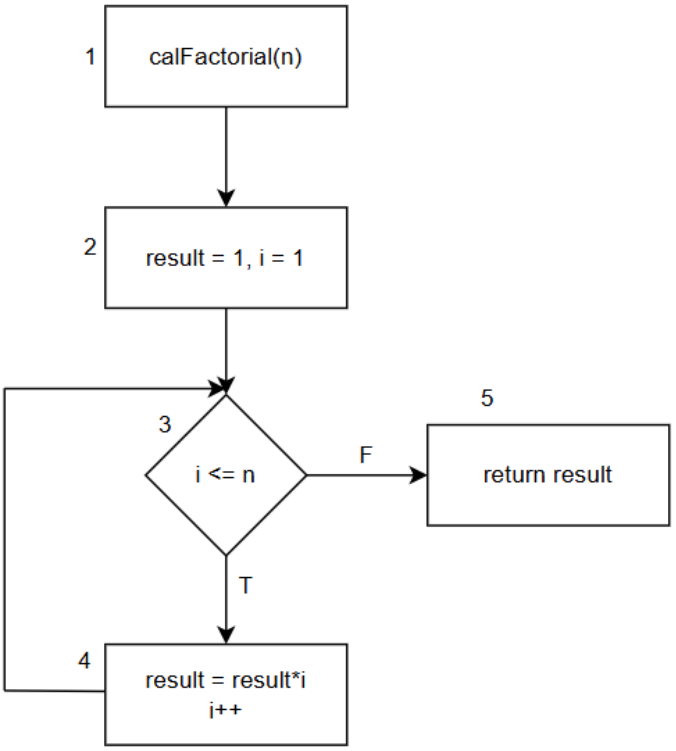
Có def(X): 1, 5 p-use(X): 3 c-use(X): 4, 6
def(Y): 1, 4 p-use(Y): 2 c-use(Y): 4, 6
Các cặp du-pair của X:
(1, 3); (1, 4); (1, 6); (5, 3); (5, 4); (5, 6)
Các cặp du-pair của Y:
(1, 2); (1, 4); (1, 6); (4, 2); (4, 4); (4, 6)

3. *Với độ do all-use (ít nhất 1 def-clear path của mọi cặp def-use)*

	Du-pair	Def-Clear Path	Complete Path	Test Case
X	(1, 3T)	1-2(T)-3(T)	1-2(T)-3(T)-4-2(F)-6	X=1, Y=1
	(1, 3F)	1-2(T)-3(F)	1-2(T)-3(F)-5-2(T)-3(T)-4-2(F)-6	X={0,1}, Y=1
	(1, 4)	1-2(T)-3(T)-4	1-2(T)-3(T)-4-2(F)-6	X=1, Y=1
	(1, 6)	1- 2(F)-6	1- 2(F)-6	X=1, Y=0
	(5, 3T)	5-2(T)-3(T)	1-2(T)-3(F)-5-2(T)-3(T)-4-2(F)-6	X={0,1}, Y=1
	(5, 3F)	5-2(T)-3(F)	1-2(T)-3(F)-5-2(T)-3(F)-5-2(T)-3(T)-4-2(F)-6	X={0,0,1}, Y=1
	(5, 4)	5-2(T)-3(T)-4	1-2(T)-3(F)-5-2(T)-3(T)-4-2(F)-6	X={0,1}, Y=1
	(5, 6)	5-2(T)-3(T)-4-2(F)-6	1-2(T)-3(F)-5-2(T)-3(T)-4-2(F)-6	X={0,1}, Y=1
Y	(1, 2T)	1-2(T)	1- 2(F)-6	X=1, Y=0
	(1, 2F)	1-2(F)	1-2(T)-3(T)-4-2(F)-6	X=1, Y=1
	(1, 4)	1-2(T)-3(T)-4	1-2(T)-3(T)-4-2(F)-6	X=1, Y=1
	(1, 6)	1- 2(F)-6	1- 2(F)-6	X=1, Y=0
	(4, 2T)	4-2(T)	1-2(T)-3(T)-4-2(T)-3(T)-4-2(F)-6	X=1, Y=2
	(4, 2F)	4-2(F)	1-2(T)-3(T)-4-2(F)-6	X=1, Y=1
	(4, 4)	4-2(T)-3(T)-4	1-2(T)-3(T)-4-2(T)-3(T)-4-2(F)-6	X=1, Y=2
	(4, 6)	4-2(F)-6	1-2(T)-3(T)-4-2(T)-3(T)-4-2(F)-6	X=1, Y=2

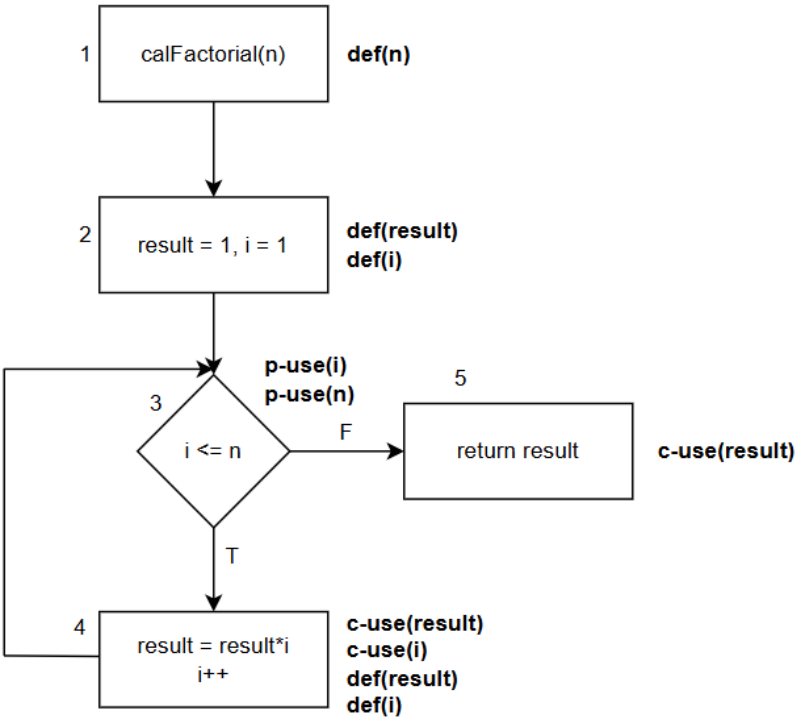
Bài 3

Control Flow Graph



Biến n:	def: 1	p-use: 3	c-use: ∅
Biến result:	def: 2, 4	p-use: ∅	c-use: 4, 5
Biến i:	def: 2, 4	p-use: 3	c-use: 4

Data Flow Graph



Bài 4

Tất cả các def-clear path của x và y

	Du-pair	Def-clear path	Du path	All p-use /some c-use	All c-use /some p-use
x	(0, 1L)	0-1L	X	X	
	(0, 1R)	0-1R	X	X	
	(0, 3)	∅			X
	(0, 4L)	0-1L-2-4L	X	X	
	(0, 4R)	0-1L-2-4R	X	X	
	(0, 5)	0-1L-2-4R-5	X		X
	(3, 1)	∅			
	(3, 3)	∅			
	(3, 4L)	3-4L	X	X	
	(3, 4R)	3-4R	X	X	
	(3, 5)	3-4R-5	X		X
y	(0, 1L)	0-1L	X	X	
	(0, 1R)	0-1R	X	X	
	(0, 4L)	0-1R-3-4L	X	X	
	(0, 4R)	0-1R-3-4R	X	X	
	(0, 6)	0-1R-3-4L-6	X		X
	(2, 1)	∅		X	
	(2, 4L)	2-4L	X	X	
	(2, 4R)	2-4R	X	X	
	(2, 6L)	2-4L-6L	X		X

	(2, 6R)	2-4L-6R	X		X
	(5, 1)	∅			
	(5, 4)	∅			
	(5, 6)	5-6	X	X	X

- Với biểu thức p-use(x,y) tại (1,3) và (4,5) là: x + y = 4 và x^2 + y^2 > 17.Lúc này đường đi (0-1-3-4-5-6) vẫn được thực hiện.

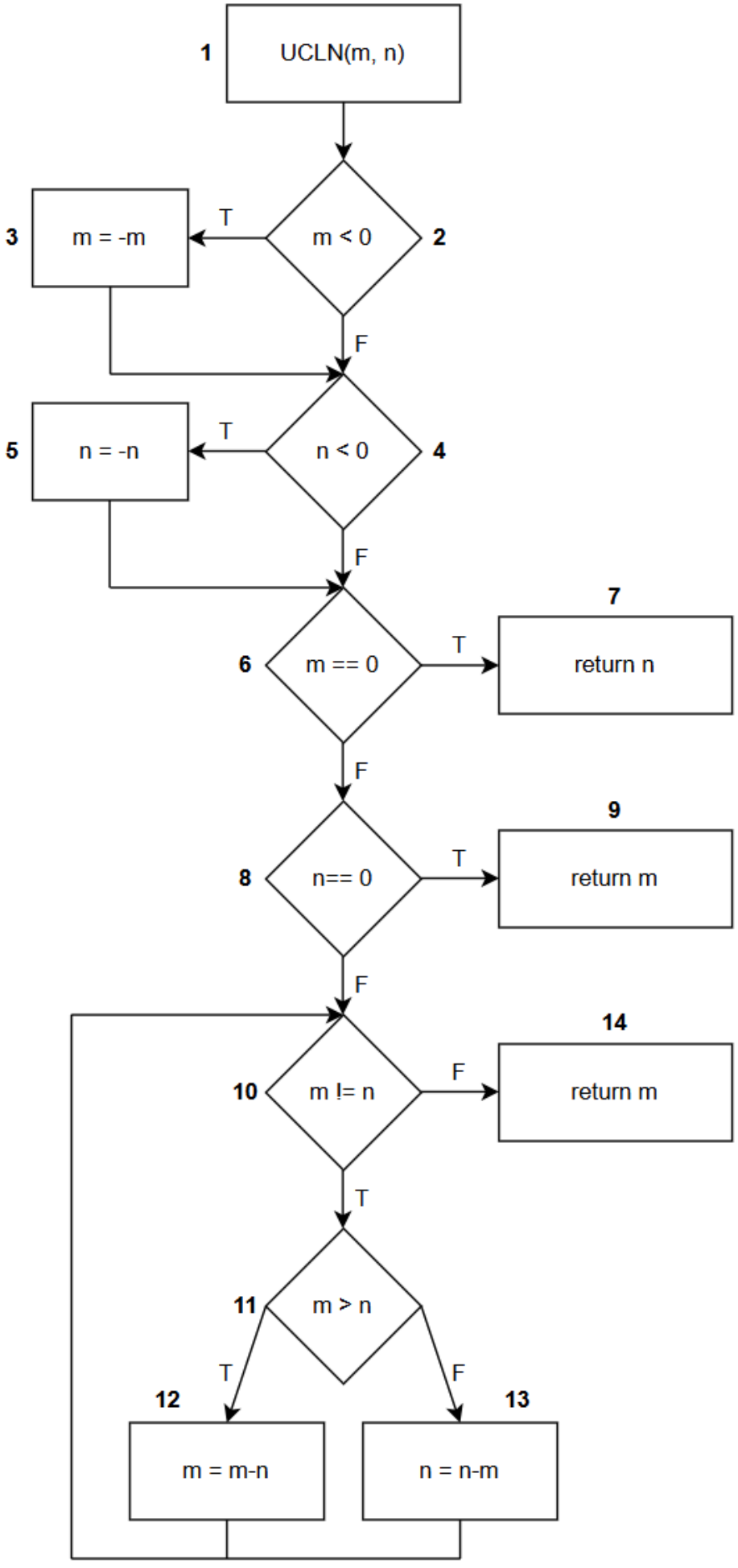
Bởi vì, không có ràng buộc x và y >= 0 nên x hoặc y có thể < 0. Vì vậy, có Test case (x = -4,y = 8) sẽ thỏa mãn điều kiện trên và đường đi được hoàn thành.

Bên cạnh đó, ở (3) có đoạn code gán giá trị cho x. Lúc này tính chất x+y=4 có thể không còn đúng nữa và do đó, điều kiện (4,5) là x^2 + y^2 > 17 không bị điều kiện (1,3) ảnh hưởng và có thể thực hiện được.

- Tại đỉnh 3, với giả sử rằng c-use(x) được thực hiện trước def(x) thì trong chính đỉnh 3 không thể tồn tại mối quan hệ def-use. Nếu ngược lại, def(x) được thực hiện trước c-use(x) thì trong đỉnh 3 vẫn tồn tại mối quan hệ def-use.

Bài 5

1. *Control Flow Graph*


1
UCLN(m, n)
m < 0
2
T
3
m = -m
F
n < 0
4
T
5
n = -n
F
m == 0
6
T
7
return n
F
n== 0
8
T
9
return m
F
m != n
10
F
14
return m
T
m > n
11
T
F
12
m = m-n
13
n = n-m

2. *Độ phủ C2: mỗi nhánh đều phải thực hiện ít nhất 1 lần*

Path	Test case
1-2T-3-4T-5-6F-8F-10T-11T-12-10T-11F-13-10F-14	m = -3, n = -2
1-2F-4F-6T-7	m = 0, n = 1
1-2F-4F-6F-8T-9	m = 1, n = 0

3. *All-def Coverage: với mỗi biến (m, n), ít nhất 1 def-clear path từ mọi lệnh def của biến tới ít nhất 1 lệnh use của nó được kiểm thử*

- Biến m:
  def: 1, 3, 12
  use: 2, 3, 6, 9, 10, 11, 12, 13, 14
- Biến n:
  def: 1, 5, 13
  use: 4, 5, 7, 8, 10, 11, 12, 13

	Du-pair	Def-Clear Path	Complete Path	Test Case
m	(1, 2)	1-2F	1-2F-4F-6T-7	m = 0, n = 1
	(3, 6)	3-4T-5-6F	1-2T-3-4T-5-6F-8F-10T-11T-12-10 T-11F-13-10F-14	m = -3, n = -2
	(12, 10)	12-10T	1-2T-3-4T-5-6F-8F-10T-11T-12-10 T-11F-13-10F-14	m = -3, n = -2
Y	(1, 4)	1-2F-4F	1-2F-4F-6T-7	m = 0, n = 1
	(5, 7)	5-6T-7	1-2F-4T-5-6T-7	m =0, n = -1
	(13, 10)	13-10F	1-2T-3-4T-5-6F-8F-10T-11T-12-10 T-11F-13-10F-14	m = -3, n = -2

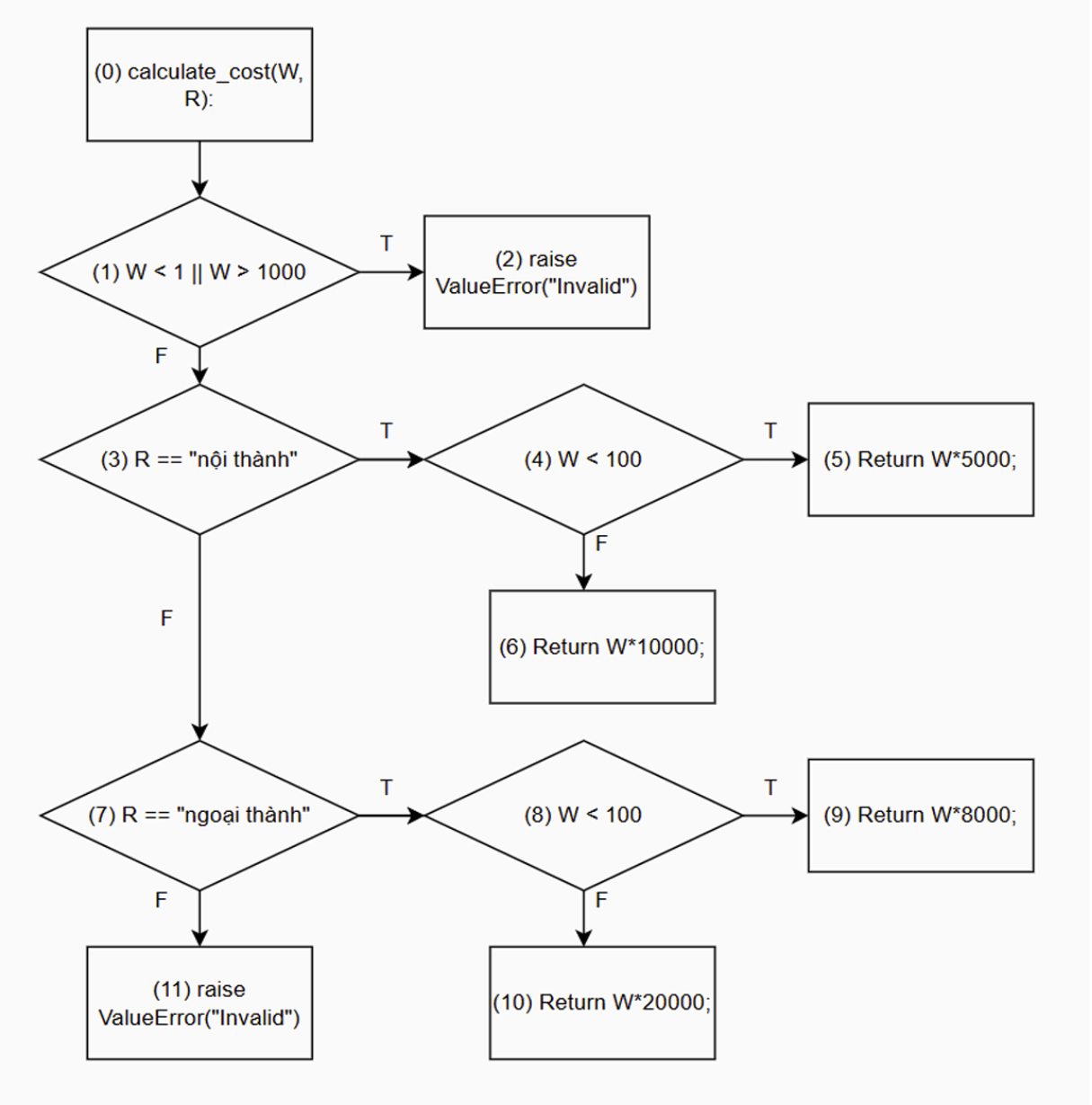
Bài 6

Code

```python
def calculate_cost(W, R):
    if W < 1 or W > 1000:
        raise ValueError("Invalid")
    if R.lower() == "nội thành":
        if W < 100:
            return int(W * 5000)
```

```
        else:
            return int(W * 10000)
    elif R.lower() == "ngoại thành":
        if W < 100:
            return int(W * 8000)
        else:
            return int(W * 20000)
    else:
        raise ValueError("Invalid")
```

Control Flow Graph



Biến W:

def: 0 p-use: 1, 4, 8 c-use: 5, 6, 9, 10

Biến R:

def: 0 p-use: 3, 7 c-use: ∅

Kiểm thử với độ phủ All-use:

	Du-pair	Def-Clear Path	Complete Path	Test Case
W	(0,1T)	0-1T	0-1T-2	W=0, R=”A”
	(0,1F)	0-1F	0-1F-3T-4T-5	W=1,R=”nội thành”
	(0,4T)	0-1F-3T-4T	0-1F-3T-4T-5	W=1,R=”nội thành”
	(0,4F)	0-1F-3T-4F	0-1F-3T-4F-6	W=100,R=”nội thành”
	(0,8T)	0-1F-3F-7T-8T	0-1F-3F-7T-8T-9	W=1,R=”ngoại thành”
	(0,8F)	0-1F-3F-7T-8F	0-1F-3F-7T-8F-10	W=100,R=”ngoại thành”
	(0,5)	0-1F-3T-4T-5	0-1F-3T-4T-5	W=1,R=”nội thành”
	(0,6)	0-1F-3T-4F-6	0-1F-3T-4F-6	W=100,R=”nội thành”
	(0,9)	0-1F-3F-7T-8T-9	0-1F-3F-7T-8T-9	W=1,R=”ngoại thành”
	(0,10)	0-1F-3F-7T-8F-10	0-1F-3F-7T-8F-10	W=100,R=”ngoại thành”
R	(0,3T)	0-1F-3T	0-1F-3T-4T-5	W=1,R=”nội thành”
	(0,3F)	0-1F-3F	0-1F-3F-7T-8T-9	W=1,R=”ngoại thành”
	(0,7T)	0-1F-3F-7T	0-1F-3F-7T-8T-9	W=1,R=”ngoại thành”
	(0,7F)	0-1F-3F-7F	0-1F-3F-7F	W=1,R=”A”

Báo cáo kết quả kiểm thử
Code được cài đặt trong file test_All_Use.py
Link Github: software-testing
Kết quả kiểm thử

ID	Input	Expected Output	Actual Output	Result
T1	0, A	Invalid	Invalid	Passed
T2	1, nội thành	5000	5000	Passed
T3	100, nội thành	1000000	1000000	Passed
T4	1, ngoại thành	8000	80000	Passed
T5	100, ngoại thành	2000000	2000000	Passed
T6	1, A	Invalid	Invalid	Passed

Vậy, chương trình pass 6/6 Test cases (100%)