

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP

MÔN HỌC: KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

BÀI TẬP: KIỂM THỬ CHỨC NĂNG

Giảng viên: ThS. Nguyễn Thu Trang

Sinh viên: Nguyễn Thị Thanh Nhân

Mã sinh viên: 22028114

Lớp: INT3117 7

HÀ NỘI - 9/2024

Mục lục

1	Mô tả bài toán	2
2	Kiểm thử bằng phương pháp kiểm thử biên	3
2.1	Xác định các giá trị	3
2.2	Thiết kế ca kiểm thử	3
3	Kiểm thử bằng phương pháp bảng quyết định	4
3.1	Xác định điều kiện và hành động	4
3.2	Lập bảng quyết định	4
3.3	Thiết kế ca kiểm thử	5
4	Thực thi kiểm thử	5
4.1	Cài đặt chương trình	5
4.2	Kiểm thử giá trị biên	6
4.3	Kiểm thử bảng quyết định	7

1 Mô tả bài toán

Cho giá bán lẻ của một sản phẩm (x) và số lượng sản phẩm mà A muốn mua (y). Tính tổng số tiền A phải trả khi mua hàng, biết phí giao hàng là 50.000đ, và các điều kiện:

- Nếu A mua trên 300 sản phẩm thì A được giảm 15%
- Nếu A mua trên 500 sản phẩm thì A được giảm 20% và miễn phí giao hàng

Đầu vào: Hai số nguyên x, y với $1000 \leq x \leq 500000$, $1 \leq y \leq 1000$

Đầu ra: Số tiền mà A phải trả.

Giá bán lẻ	Số lượng sản phẩm	Chi phí
$1000 \leq x \leq 500000$	$1 \leq y < 300$	$\text{total} = x \times y + 50000$
	$300 \leq y < 500$	$\text{total} = x \times y \times 85\% + 50000$
	$500 \leq y \leq 1000$	$\text{total} = x \times y \times 80\%$

2 Kiểm thử bằng phương pháp kiểm thử biên

2.1 Xác định các giá trị

Ta sử dụng phương pháp kiểm thử giá trị biên mạnh nên cần xác định 7 giá trị với mỗi biến:

x, y với $1000 \leq x \leq 500000, 1 \leq y \leq 1000$	
$\min_x = 1000$	$\min_y = 1$
$\max_x = 500000$	$\max_y = 1000$
$\min_{x+} = 1001$	$\min_{y+} = 2$
$\max_{x-} = 499999$	$\max_{y-} = 999$
$\min_{x-} = 999$	$\min_{y-} = 0$
$\max_{x+} = 500001$	$\max_{y+} = 1001$
$\text{nom}_x = 250000$	$\text{nom}_y = 500$

2.2 Thiết kế ca kiểm thử

STT	x	y	Expected output
1	1000	500	400000
2	500000	500	200000000
3	1001	500	400400
4	499999	500	199999600
5	999	500	invalid input
6	500001	500	invalid input
7	250000	1	300000
8	250000	1000	200000000
9	250000	2	550000
10	250000	999	199800000
11	250000	0	invalid input
12	250000	1001	invalid input
13	250000	500	100000000

3 Kiểm thử bằng phương pháp bảng quyết định

3.1 Xác định điều kiện và hành động

Hai số nguyên x, y với $1000 \leq x \leq 500000$, $1 \leq y \leq 1000$

C1: $x < 1000 \cup x > 500000$

C2: $1000 \leq x \leq 500000$

C3: $y < 1 \cup y > 1000$

C4: $1 \leq y < 300$

C5: $300 \leq y < 500$

C6: $500 \leq y \leq 1000$

E1: invalid input

E2: $\text{total} = x \times y + 50000$

E3: $\text{total} = x \times y \times 85\% + 50000$

E4: $\text{total} = x \times y \times 80\%$

3.2 Lập bảng quyết định

		R1	R2	R3	R4	R5	R6	R7	R8
Điều kiện	$x < 1000 \cup x > 500000$	T	T	T	T	F	F	F	F
	$1000 \leq x \leq 500000$	-	-	-	-	T	T	T	T
	$y < 1 \cup y > 1000$	T	F	F	F	T	F	F	F
	$1 \leq y < 300$	-	T	F	F	-	T	F	F
	$300 \leq y < 500$	-	-	T	F	-	-	T	F
	$500 \leq y \leq 1000$	-	-	-	T	-	-	-	T
Hành động	invalid input	x	x	x	x	x			
	$\text{total} = x \times y + 50000$						x		
	$\text{total} = x \times y \times 85\% + 50000$							x	
	$\text{total} = x \times y \times 80\%$								x

3.3 Thiết kế ca kiểm thử

STT	x	y	Expected output
1	950	2000	invalid input
2	850	200	invalid input
3	600000	400	invalid input
4	800000	600	invalid input
5	10000	3000	invalid input
6	20000	150	3050000
7	30000	450	11525000
8	40000	650	20800000

4 Thực thi kiểm thử

4.1 Cài đặt chương trình

```
1 def calculate_total_cost(x, y):
2     shipping = 50000
3
4     if not (1000 <= x <= 500000) or not (1 <= y <= 1000):
5         return "invalid input"
6
7     if 1 <= y < 300: # No discount
8         return x * y + shipping
9     elif 300 <= y < 500: # 15% discount
10        return x * y * 0.85 + shipping
11    elif 500 <= y <= 1000: # 20% discount and free shipping
12        return x * y * 0.80
```

4.2 Kiểm thử giá trị biên

```
1 from calculate import calculate_total_cost
2
3 import pytest
4
5 # Test cases for boundary value testing
6 @pytest.mark.parametrize(
7     "x, y, expected_output",
8     [
9         (1000, 500, 400000),
10        (500000, 500, 200000000),
11        (1001, 500, 400400),
12        (499999, 500, 199999600),
13        (999, 500, "invalid input"),
14        (500001, 500, "invalid input"),
15        (250000, 1, 300000),
16        (250000, 1000, 200000000),
17        (250000, 2, 550000),
18        (250000, 999, 199800000),
19        (250000, 0, "invalid input"),
20        (250000, 1001, "invalid input"),
21        (250000, 500, 100000000),
22    ]
23 )
24
25 def test_calculate_total_cost(x, y, expected_output):
26     result = calculate_total_cost(x, y)
27     print(f"Test case with x={x}, y={y}: expected {expected_output}, got {result}")
28     assert result == expected_output
29
30 if __name__ == "__main__":
31     pytest.main()
```

STT	x	y	Expected output	Actual output	Result
1	1000	500	400000	400000	Pass
2	500000	500	200000000	200000000	Pass
3	1001	500	400400	400400	Pass
4	499999	500	199999600	199999600	Pass
5	999	500	invalid input	invalid input	Pass
6	500001	500	invalid input	invalid input	Pass
7	250000	1	300000	300000	Pass
8	250000	1000	200000000	200000000	Pass
9	250000	2	550000	550000	Pass
10	250000	999	199800000	199800000	Pass
11	250000	0	invalid input	invalid input	Pass
12	250000	1001	invalid input	invalid input	Pass
13	250000	500	100000000	100000000	Pass

4.3 Kiểm thử bảng quyết định

```
1 from calculate import calculate_total_cost
2
3 import pytest
4
5 # Test cases for decision table testing
6 @pytest.mark.parametrize(
7     "x, y, expected_output",
8     [
9         (999, 2000, "invalid input"),
10        (850, 200, "invalid input"),
11        (600000, 400, "invalid input"),
12        (800000, 600, "invalid input"),
13        (10000, 3000, "invalid input"),
14        (20000, 150, 3050000),
15        (30000, 450, 11525000),
16        (40000, 650, 20800000),
17    ]
18 )
19
20 def test_calculate_total_cost(x, y, expected_output):
21     result = calculate_total_cost(x, y)
22     print(f"Test case with x={x}, y={y}: expected {expected_output}, got {result}")
23     assert result == expected_output
24
25 if __name__ == "__main__":
26     pytest.main()
```

STT	x	y	Expected output	Actual output	Result
1	950	2000	invalid input	invalid input	Pass
2	850	200	invalid input	invalid input	Pass
3	600000	400	invalid input	invalid input	Pass
4	800000	600	invalid input	invalid input	Pass
5	10000	3000	invalid input	invalid input	Pass
6	20000	150	3050000	3050000	Pass
7	30000	450	11525000	11525000	Pass
8	40000	650	20800000	20800000	Pass