**2.1. Giả sử cần kiểm thử chương trình tính chu vi tam giác (P = a+b+c) với đầu vào là 3 số nguyên a, b, c thuộc [1,100]. Hãy thiết kế các ca kiểm thử (test case) tối thiểu cho chương trình trên áp dụng phương pháp kiểm thử giá trị biên yếu (BVA) và phương pháp phân lớp tương đương (ECT) cho miền giá trị đầu vào và đầu ra.**

### int tinhChuViTamGiac(int a, int b, int c) {

### if (a + b > c && a + c > b && b + c > a) {

### return a + b + c;

### } else {

### return -1;

### }

### }

#### **1. Phương pháp kiểm thử giá trị biên yếu (BVA)**

Kiểm thử các giá trị ở biên giới của miền giá trị đầu vào.

* **Biên dưới của miền giá trị hợp lệ:**
  + **Ví dụ:** a = 1, b = 1, c = 1 (tam giác đều với cạnh nhỏ nhất)
  + **Expected Output:** 3
* **Biên trên của miền giá trị hợp lệ**
  + **Ví dụ:** a = 100, b = 100, c = 100 (tam giác đều với cạnh lớn nhất)
  + **Expected Output:** 300
* **Biên kết hợp:** Một cạnh ở biên trên và hai cạnh ở biên dưới.
  + **Ví dụ:** a = 1, b = 1, c = 100
  + **Expected Output:** -1 (không phải là tam giác)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ca kiểm thử** | **R1** | **R2** | **R3** | **R4** | **R5** |
| **a** | 1 | 100 | 0 | 70 | 80 |
| **b** | 1 | 100 | 50 | 102 | 90 |
| **c** | 1 | 100 | 100 | 80 | 110 |
| **Ghi chú** | Biên dưới hợp lệ | Biên trên hợp lệ | a ngoài biên dưới | b ngoài biên trên | c ngoài biên trên |
| **Expected Output** | Chu vi = 3 | Chu vi = 300 | Không hợp lệ | Không hợp lệ | Không hợp lệ |

#### **2. Phương pháp phân lớp tương đương (ECT)**

Phân lớp các giá trị thành các nhóm tương đương và chọn các giá trị đại diện từ mỗi nhóm.

* **Lớp hợp lệ:** Các giá trị mà ba số a, b, c có thể tạo thành một tam giác.
  + **Ví dụ:** a = 3, b = 4, c = 5 (tam giác vuông)
  + **Expected Output:** 12
* **Lớp không hợp lệ:** Các giá trị mà ba số a, b, c không thể tạo thành một tam giác.
  + **Ví dụ:** a = 1, b = 2, c = 3
  + **Expected Output:** -1
* **Lớp hợp lệ khác:** Tam giác có ba cạnh bằng nhau.
  + **Ví dụ:** a = 5, b = 5, c = 5
  + **Expected Output:** 15

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ca kiểm thử** | **R1** | **R2** | **R3** | **R4** | **R5** |
| **a** | 3 | 1 | 5 | 25 | -1 |
| **b** | 4 | 2 | 5 | 50 | 0 |
| **c** | 5 | 3 | 5 | 150 | 101 |
| **Ghi chú** | Tất cả các giá trị hợp lệ | a không hợp lệ | Tất cả các giá trị hợp lệ | c không hợp lệ | Tất cả không hợp lệ |
| **Expected Output** | Chu vi = 12 | Không hợp lệ | Chu vi = 12 | Không hợp lệ | Không hợp lệ |

**2.2. Hãy thiết kế các ca kiểm thử để phủ hết tất cả lộ trình và các kiểm thử để kiểm thứ hai vòng lặp While của đoạn chương trình sau:**

**void mergeTwoArrays(int a[], int n, int b[], int m, int c[])**

**{**

**int min, i = 0, j = 0;**

**if(n > m)**

**min = m;**

**else**

**min = n;**

**for(i = 0; i < min; i++, j += 2)**

**{**

**c[j] = a[i];**

**c[j + 1] = b[i];**

**}**

**while(i < n)**

**{**

**c[j++] = a[i++];**

**}**

**while(i < m)**

**{**

**c[j++] = b[i++];**

**}**

**}**

**#Kiểm thử các lộ trình**

**Test Case 1: n = 3, m=3**

* **Mô tả:** Cả hai mảng a và b đều có độ dài bằng nhau và vòng lặp for sẽ thực thi đúng một lần.
* **Input:**
  + a[] = {1, 2, 3}
  + b[] = {4, 5, 6}
  + n = 3, m = 3
* **Expected Output:**
  + c[] = {1, 4, 2, 5, 3, 6}

**Test Case 2: n > m**

* **Mô tả:** Mảng a có độ dài lớn hơn mảng b. Vòng lặp while đầu tiên sẽ thực thi để thêm các phần tử còn lại của a vào c.
* **Input:**
  + a[] = {1, 2, 3, 4}
  + b[] = {5, 6}
  + n = 4, m = 2
* **Expected Output:**
  + c[] = {1, 5, 2, 6, 3, 4}

**Test Case 3: n < m**

* **Mô tả:** Mảng b có độ dài lớn hơn mảng a. Vòng lặp while thứ hai sẽ thực thi để thêm các phần tử còn lại của b vào c.
* **Input:**
  + a[] = {1, 2}
  + b[] = {3, 4, 5}
  + n = 2, m = 3
* **Expected Output:**
  + c[] = {1, 3, 2, 4, 5}

**Test Case 4: n = 0, m = 2**

* **Mô tả:** Mảng a rỗng, mảng b có hai phần tử.
* **Input:**
  + a[] = {}
  + b[] = {3, 4}
  + n = 0, m = 2
* **Expected Output:**
  + c[] = {3, 4}

**Test Case 5: n = 2, m = 0**

* **Mô tả:** Mảng a có hai phần tử, mảng b rỗng.
* **Input:**
  + a[] = {1, 2}
  + b[] = {}
  + n = 2, m = 0
* **Expected Output:**
  + c[] = {1, 2}

**Test Case 6: n = 0, m = 0**

* **Mô tả:** Cả hai mảng đều rỗng. Không có vòng lặp nào thực thi.
* **Input:**
  + a[] = {}
  + b[] = {}
  + n = 0, m = 0
* **Expected Output:**
  + c[] = {}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ca kiểm thử** | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** |
| **n** | 3 | 4 | 2 | 0 | 2 | 0 |
| **m** | 3 | 2 | 3 | 2 | 0 | 0 |
| **a** | [1, 2, 3] | [1, 2, 3, 4] | [1, 2] | [] | [1, 2] | [] |
| **b** | [4, 5, 6] | [5, 6] | [3, 4, 5] | [3, 4] | [] | [] |
| **Expected Output** | [1, 4, 2, 5, 3, 6] | [1, 5, 2, 6, 3, 4] | [1, 3, 2, 4, 5, 6] | [3, 4] | [1, 2] | [] |
| **Ghi chú** | n = m, cả hai mảng có cùng số lượng phần tử | n > m, mảng a dài hơn mảng b | n < m, mảng b dài hơn mảng a | n = 0, mảng a rỗng | m = 0, mảng b rỗng | n = 0 và m = 0, cả hai mảng đều rỗng |

**#Kiểm thử vòng lặp While**

**Test Case 1: n = 4, m=2**

* **Mô tả:** Vòng lặp while thứ nhất hoạt động, vòng lặp thứ hai không hoạt động
* **Input:**
  + a[] = {1, 2, 3, 4}
  + b[] = {5, 6}
  + n = 4, m = 2
* **Expected Output:**
  + c[] = {1, 5, 2, 6, 3, 4}

**Test Case 2: n = 2, m=4**

* **Mô tả:** Vòng lặp while thứ nhất không hoạt động, vòng lặp thứ hai hoạt động
* **Input:**
  + a[] = {1, 2}
  + b[] = {3, 4, 5, 6}
  + n = 2, m = 4
* **Expected Output:**
  + c[] = {1, 3, 2, 4, 5, 6}