

Ejercicios Resueltos de Pruebas de Software

Ejercicio1 : Dado el siguiente fragmento de programa en java:

```

Import java.io.*;

Public class Maximo
{
    public static void main (String args[]) throws IOException
    {
        BufferedReader  entrada  =  new  BufferedReader  (new
        InputStreamReader(System.in));

        Int x,y,z,max;

        System.out.println("Introduce x,y,z: ");
        x = Integer.parseInt (entrada.readLine());
        y = Integer.parseInt (entrada.readLine());
        z = Integer.parseInt (entrada.readLine());

        if (x>y && x>z)
            max = x;
        else
            if (z>y)
                max = z;
            else
                max = y;
        System.out.println ("El máximo es "+ max);
    }
}

```

Se pide

a. Calcular la complejidad ciclomática de McCabe $V(G)$

Para calcular la complejidad ciclomática de McCabe, lo primero que tenemos que hacer es dibujar el grafo de flujo. Para esto seguiremos los siguientes pasos:

1. Señalamos en el código los pasos para dibujar el grafo de flujo

```

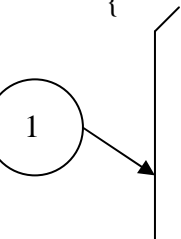
Import java.io.*;

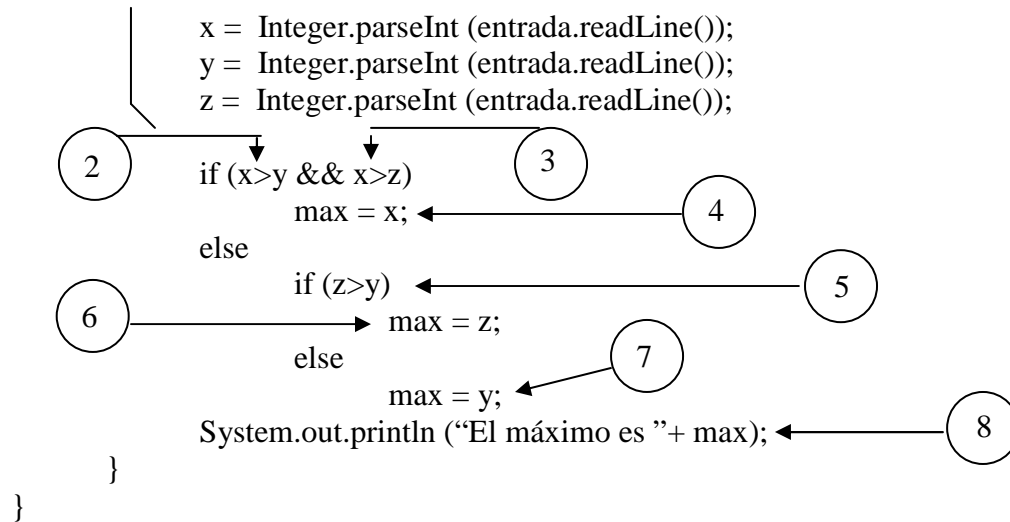
Public class Maximo
{
    public static void main (String args[]) throws IOException
    {
        BufferedReader  entrada  =  new  BufferedReader  (new
        InputStreamReader(System.in));

        Int x,y,z,max;

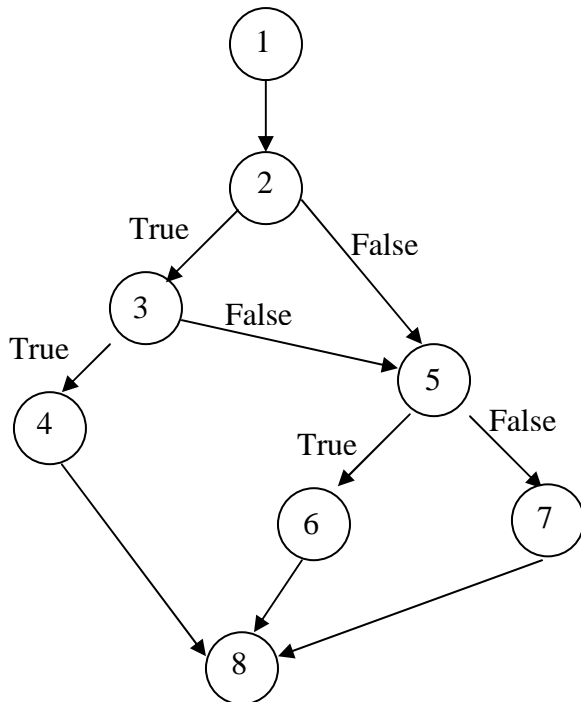
        System.out.println("Introduce x,y,z: ");
    }
}

```





2. Dibujamos el grafo de flujo , calculamos la complejidad ciclomática y determinamos los caminos independientes:



Calculamos la complejidad ciclomática de McCabe :

$$V(G) = a - n + 2 = 10 - 8 + 2 = 4$$

$$V(G) = r = 4$$

$$V(G) = c + 1 = 3 + 1 = 4$$

Por lo tanto tendremos cuatro caminos independientes, que mirando el grafo de flujo deducimos serán los siguientes:

- Camino 1 ➔ 1 - 2 - 3 - 4 - 8
- Camino 2 ➔ 1 - 2 - 3 - 5 - 6 - 8

- Camino 3 → 1 – 2 – 5 – 6 – 8
- Camino 4 → 1 – 2 – 5 – 7 – 8

b. Definir conjuntos de pruebas mínimo para alcanzar los siguientes criterios de cobertura:

- **Cobertura de sentencias**

Se trata de ejecutar con los casos de prueba cada sentencia e instrucción al menos una vez.

En este caso con ejecutar los caminos 1, 2 y 4 nos vale:

Camino	Características	Caso de Prueba		
		x	y	z
Camino 1	$x > y, x > z$	10	3	3
Camino 2	$y < x < z$	5	2	10
Camino 4	$x < y, z < y$	5	10	5

- Caso de prueba 1 (Camino 1). Ejecutaremos un caso en el que $x > y$ y $x > z$, como por ejemplo $x = 10, y = 3$ y $z = 3$.
- Caso de prueba 2(Camino 2).Ejecutamos un caso en el que $y < x < z$, como por ejemplo $(x,y,z)=(5,2,10)$
- Caso de prueba 3 (Camino 4). Ejecutamos un caso en el que $x < y$ y $z < y$, como por ejemplo $(x,y,z)=(5,10,5)$

- **Cobertura de decisiones**

Escribimos los casos suficientes para que cada condición tenga al menos una resultado verdadero y otro falso. Utilizando los mismos caminos y casos de prueba que en la cobertura de sentencias cubriremos también en este caso la cobertura de decisiones:

Camino	Características	Caso de Prueba		
		x	y	z
Camino 1	$x > y, x > z$	10	3	3
Camino 2	$y < x < z$	5	2	10
Camino 4	$x < y, z < y$	5	10	5

- **Cobertura de condiciones**

Se trata de escribir los casos suficientes para que cada condición de cada decisión adopte el valor verdadero y el falso al menos una vez. Los casos de prueba en este caso serán los mismos que en la cobertura de decisiones.

- **Cobertura de decisión /condición**

Es el cumplimiento de la cobertura de condiciones y de decisiones. Elegiremos los caminos 1,2 y 4, con los casos de prueba vistos anteriormente.

- **Criterio de condición múltiple**

Si tenemos decisiones multicondicionales las descompondremos en decisiones unicondicionales, ejecutando todas las combinaciones posibles de resultados.

Tenemos la decisión multicondicional $x > y \ \&\& \ x > z$ y la decisión unicondicional $z > y$. Combinándolas nos damos cuenta que para que se cumpla el criterio de condición múltiple debemos ejecutar los cuatro caminos independientes:

Camino	Características	Caso de Prueba		
		x	y	z
Camino 1	$x > y \ , \ x > z$	10	3	3
Camino 2	$y < x < z$	5	2	10
Camino 3	$x < y < z$	2	5	8
Camino 4	$x < y \ , \ z < y$	5	10	5

Ejercicio 2: *Dado el siguiente fragmento de programa en java:*

If (a>1) and (b>5) and (c<2) then

x=x+1;

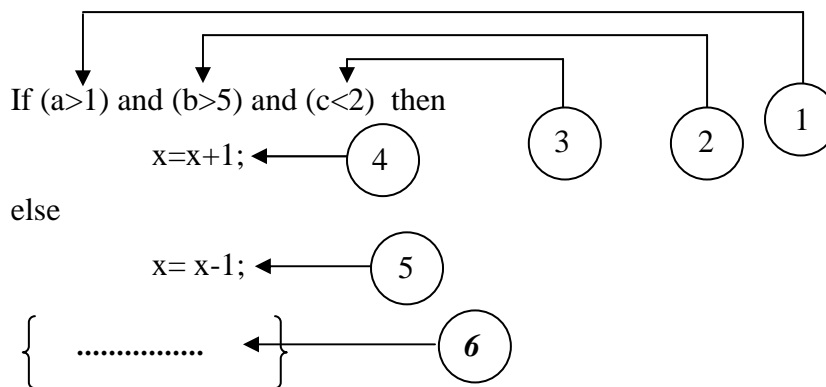
else

x= x-1;

¿Qué criterios de cobertura satisfacen los siguientes casos de prueba {a=0, b=11, c=1}, {a=4,b=4,c=4}, {a=2, b=6, c=0}?Explicar cada uno de ellos

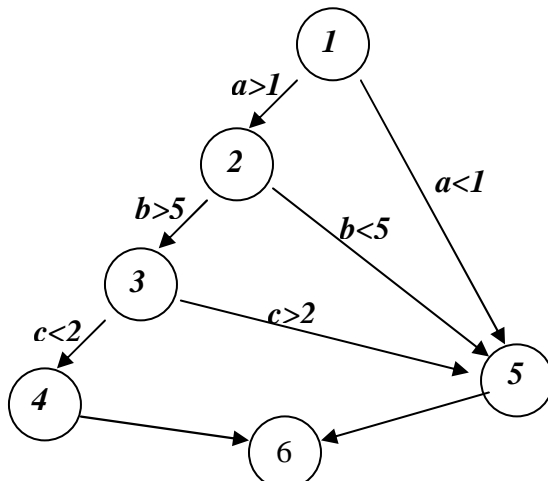
Calculamos el Grafo de flujo y la complejidad ciclomática, de la misma manera que en el ejercicio anterior. De esta manera veremos con más facilidad cuales son los pasos que se ejecutan con los casos de prueba que nos dan.

1. Señalamos en el fragmento de código los nodos:



Incluimos un sexto nodo que hace referencia al código que supuestamente viene después (en el enunciado no lo determina). De esta manera podremos calcular la complejidad ciclomática sin errores.

2. Dibujamos el grafo de flujo:



3. Calculamos la complejidad ciclomática:

$$V(G) = 8 - 6 + 2 = 4 \text{ caminos independientes}$$

Camino 1 → 1 – 5 – 6

Camino 2 → 1- 2 – 5 – 6

Camino 3 → 1 – 2 – 3 – 5 – 6

Camino 4 → 1 – 2 – 3 – 4 – 6

Con el caso de prueba {a=0, b=11, c=1} ejecutamos el primer camino independiente.

Con el caso de prueba {a=4, b=4, c=4} ejecutamos el segundo camino independiente.

Con el caso de prueba {a=2, b=6, c=0} ejecutamos el cuarto camino independiente.

- **Cobertura de sentencias:**

Se trata de ejecutar con los casos de prueba cada sentencia e instrucción al menos una vez. Ejecutando los casos de prueba que nos dan en el enunciado ejecutaremos cada instrucción al menos una vez: si nos fijamos en los caminos independientes nos daremos cuenta de que el flujo de ejecución pasa por todos los nodos del código.

- **Cobertura de decisiones:**

Escribimos los casos suficientes para que cada condición tenga al menos una resultado verdadero y otro falso. En nuestro caso bastaría con ejecutar el interior del if y el interior del else, sin tener en cuenta las condiciones. Cuando tenemos decisiones multicondicionales puede ejecutarse el bloque else con diferentes combinaciones estas.

En este caso se cumple la cobertura de condiciones ya que se ejecutan el nodo 4 y 5, que son las sentencias del interior del if y del else respectivamente.

- **Cobertura de condiciones:**

Se trata de escribir los casos suficientes para que cada condición de cada decisión adopte el valor verdadero y el falso al menos una vez. En este caso a y b adoptan el valor verdadero y el valor falso al menos una vez.

En el caso de c, comprobamos que en el caso de prueba {a=2, b=6, c=0} adopta el valor verdadero, pero no encontramos el caso de prueba en el que se ejecute el valor falso. En el caso de prueba {a=4, b=4, c=4} pese a tener un valor falso no podemos

considerarlo puesto que en este mismo caso de prueba b también adopta un valor falso y se ejecuta antes. De esta manera $c=4$ no llegaría nunca a ejecutarse.

Por este motivo concluimos que no se cumple la cobertura de condiciones.

- ***Cobertura de decisión/condición***

Es el cumplimiento de la cobertura de condiciones y de decisiones. Evidentemente en este caso no se cumple, puesto que hemos dicho que no se cumple la cobertura de condiciones.

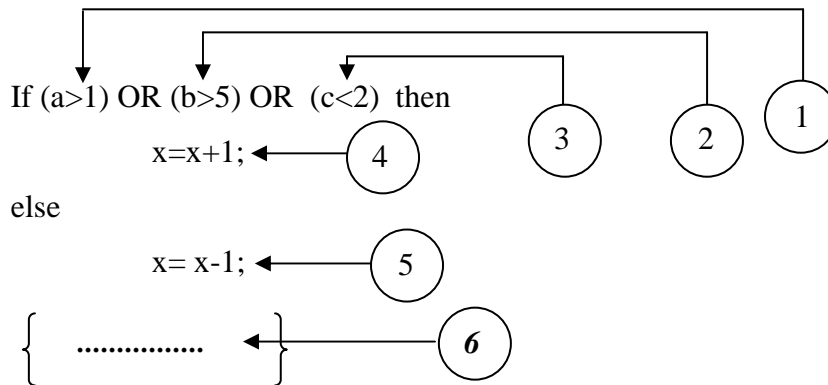
- ***Criterio de condición múltiple***

Cuando nos encontramos ante una decisión multicondicional según este criterio es preciso descomponerla en decisiones unicondicionales, ejecutando todas las combinaciones posibles de resultados.

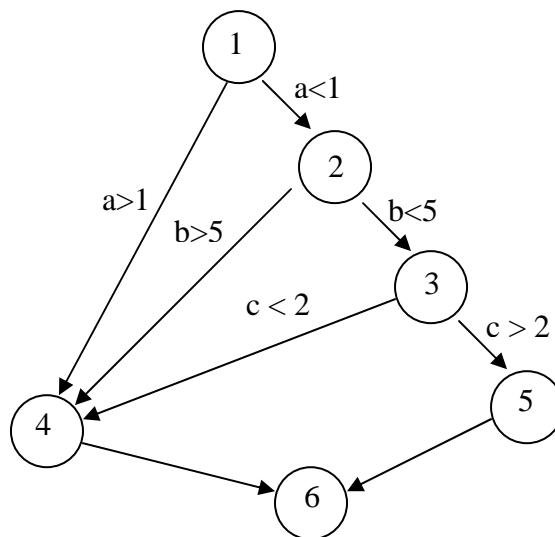
En este caso para que se cumpla este criterio deberían de ejecutarse los cuatro caminos independientes que hemos calculado. De esta manera, tendríamos todas las combinaciones posibles de resultados. Al ejecutarse solo tres de ellos podemos decir que no se cumple el criterio de condición múltiple.

Ejercicio 3: Dado el mismo fragmento de código del ejercicio anterior con dos cláusulas OR, calcular el número mínimo de casos de prueba que es necesario para que se cumpla la cobertura de condiciones.

1. Señalamos en el código los nodos :



2. Dibujamos el grafo de flujo. En este caso al trarse de cláusulas OR en vez de AND el dibujo cambia notablemente:



3. Calculamos la complejidad ciclomática:

$$V(G) = 8 - 6 + 2 = 4 \text{ caminos independientes}$$

Camino 1 ➔ 1 - 4

Camino 2 ➔ 1 - 2 - 4

Camino 3 ➔ 1 – 2 – 3 – 4

Camino 4 ➔ 1 – 2 – 3 – 5 – 6

Para que se cumpla el criterio de cobertura de condiciones, es decir, que cada condición de cada decisión tome el valor falso y verdadero al menos una vez, deberán ejecutarse los cuatro caminos críticos. Es decir, el número mínimo de pruebas que deberemos realizar será cuatro.