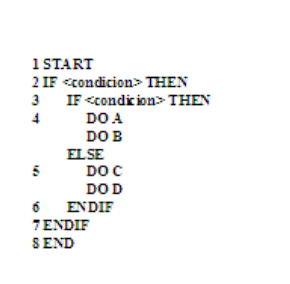
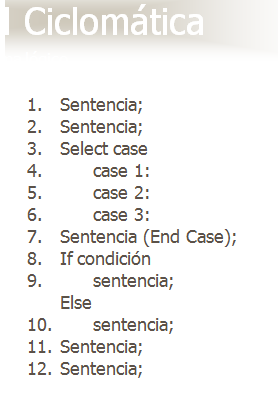
***Ejercicio 1*** :**Hallar el grafo del algoritmo y la complejidad ciclomática.**

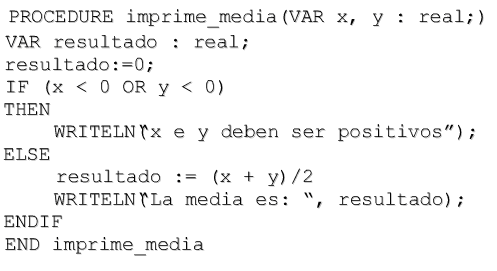


***Ejercicio 2*** :**Hallar el grafo del algoritmo y la complejidad ciclomática.**

**END IF**



***Ejercicio 3*** :**Hallar el grafo del algoritmo y la complejidad ciclomática (ciclo3.dia).**



***Ejercicio 4*** :**Hallar el grafo del algoritmo y la complejidad ciclomática**

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

x=x+1;

else

x= x-1;

***Ejercicio 5*** :***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.***

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

x=x+1;

else

x= x-1;

***Ejercicio 6*** :***Dado el siguienteprograma en java Calcular la complejidad ciclomática***

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

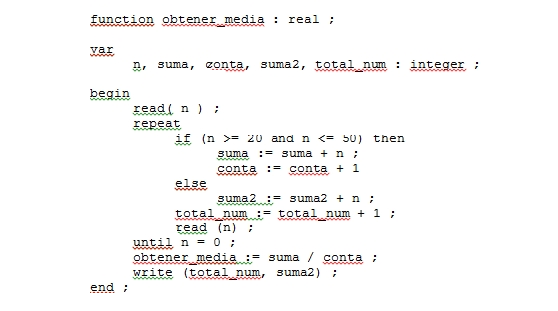
max = y;

System.out.println (“El máximo es ”+ max);

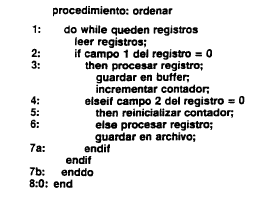
}

}

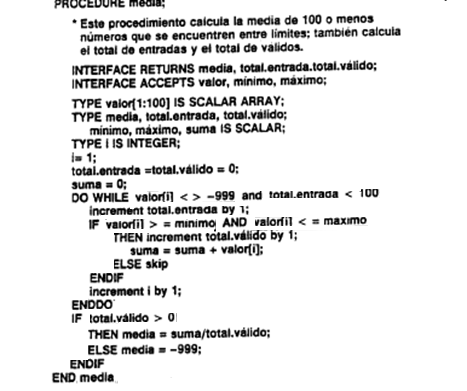
***Ejercicio 7***:**Hallar el grafo del algoritmo,V(G) y los caminos independientes**



***Ejercicio 8*** :



***Ejercicio 9*** :



Clarificando un poco la parte el DO, el código se podría poner también así

Do while …..

Increment

If valor[i] >= minimo AND valor[i] <=maximo THEN

Increment total.valido by 1;

Suma=suma + valor[i];

Else

Skip

End if

Increment i by 1;

End do

If total.valido >0 THEN

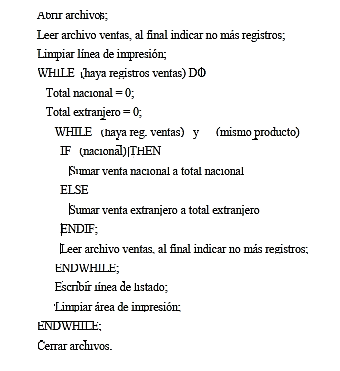
Media = suma / total.valido;

Else

Media= -999;

End if

***Ejercicio 10*** :



***EJERCICIO 11:***Dado el siguiente código:

IF (X + Y + Z)/3 = X THEN

Print(“X, Y, Z son iguales”)

ELSE

Print(“X, Y, Z no son iguales”)

END IF

Hay que proponer dos casos de prueba (especificándolos en la tabla de abajo) para conseguir la cobertura de decisiones y que, sin embargo, no permita descubrir que este código no funciona correctamente.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Num. Caso | X | Y | Z | Resultado esperado | Resultado conseguido |
|  |  |  |  |  |  |

Propón también un caso de prueba documentándolo en la tabla que permita descubrir el defecto de lógica que contiene el código.