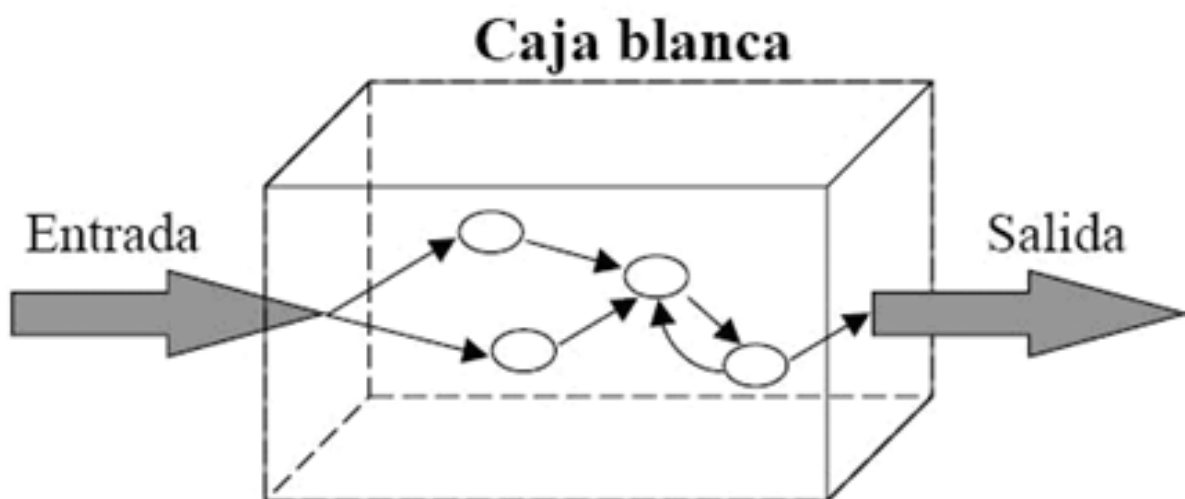


## Ejercicios de Pruebas de Caja Blanca



# Ejercicio - 1

Código:

```
1 if (a > 1 and b > 5 and c < 2) then
2 x = x + 1
3 else
4 x = x - 1
5 end
```

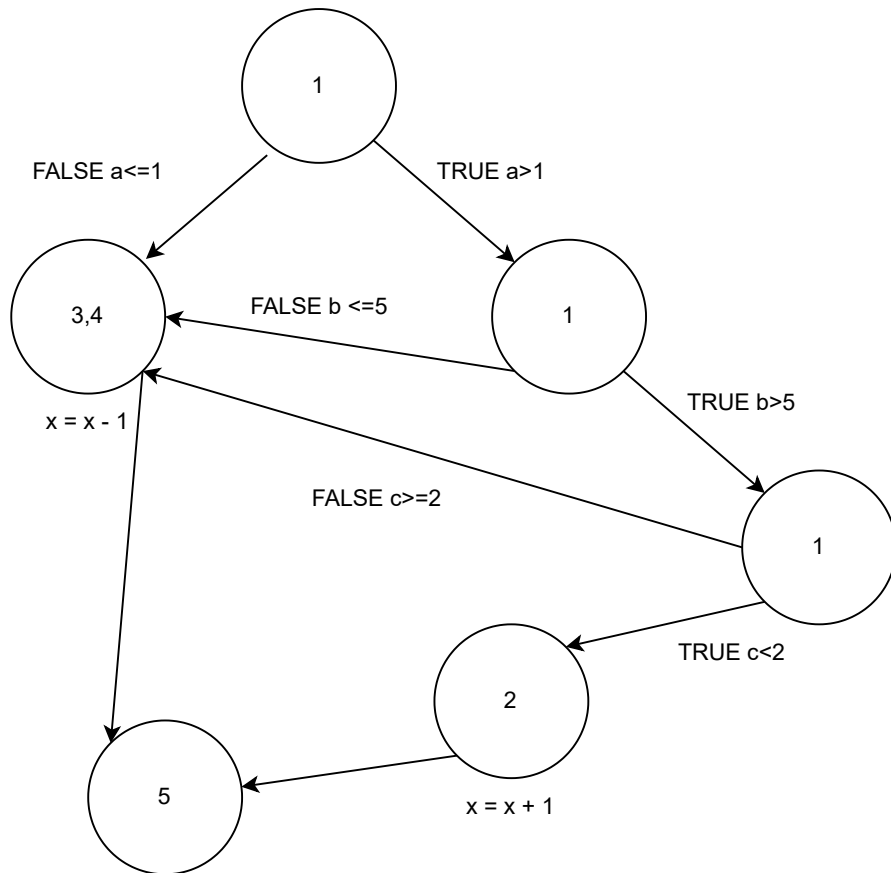
Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$ , Hay 8 Aristas y 6 Nodos

ó

$V(G) = nps + 1 = 4$ , Hay 3 Nodos con Dos Aristas

Programa Simple Sin Mucho Riesgo.



## Ejercicio - 2

Código:

```
1 if (a > 1 or b > 5 or c < 2) then
2   x = x + 1
3 else
4   x = x - 1
5 end
```

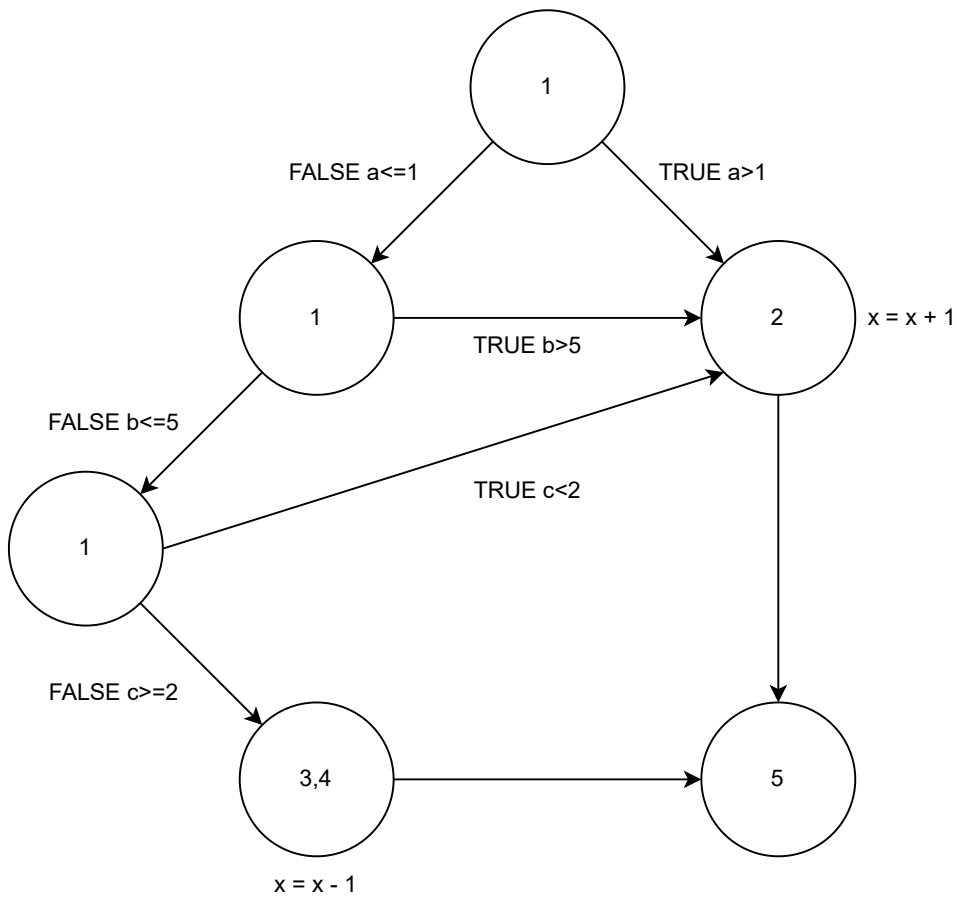
Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$  Hay 8 Aristas y 6 Nodos.

ó

$V(G) = nps + 1 = 4$  Hay 3 Nodos con 2 Aristas.

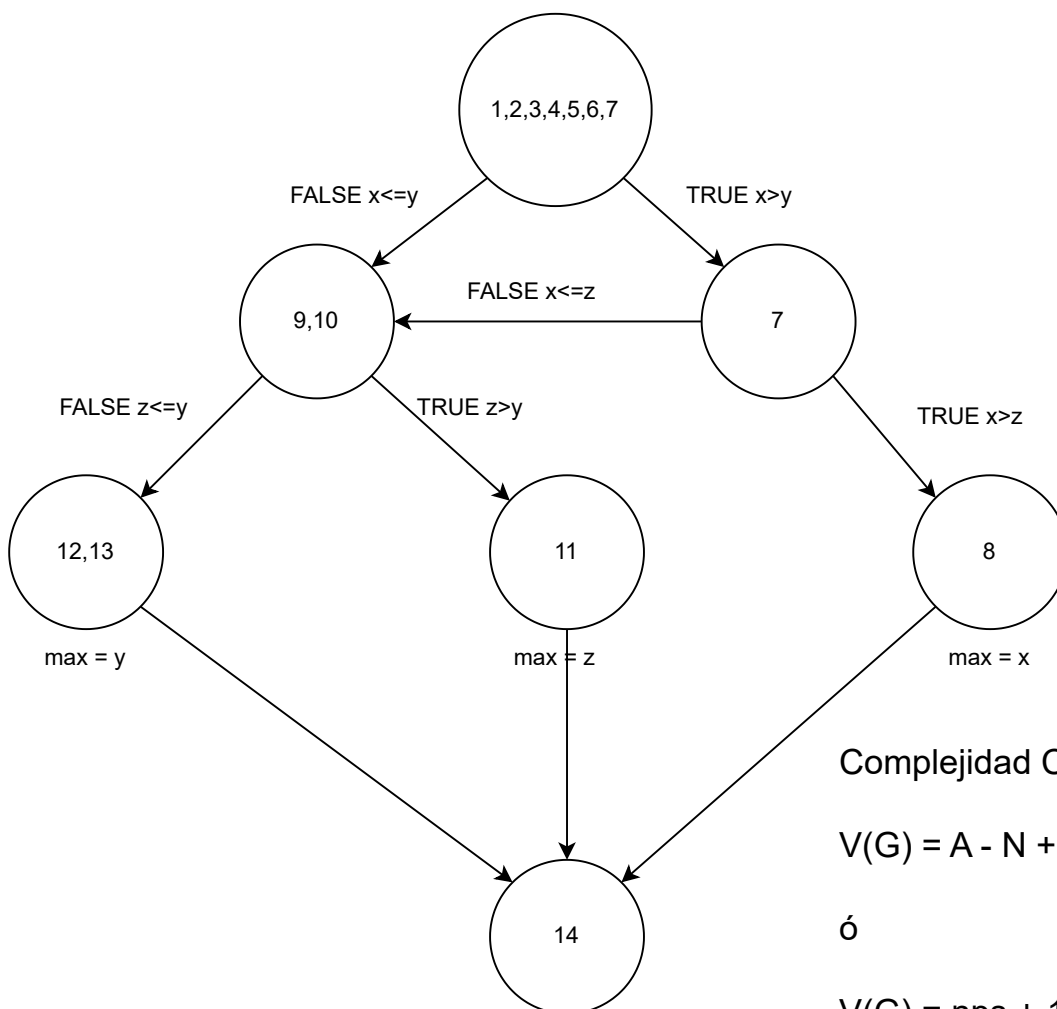
Programa Simple sin Mucho Riesgo.



# Ejercicio - 3

```
Import java.io.*;

Public class Maximo
{
    public static void main (String args[]) throws IOException
    {
1      BufferedReader entrada = new BufferedReader (new InputStreamReader (System.in));
2
3      int x,y,z,max;
4
5      System.out.println("Introduce x,y,z: ");
6      x = Integer.parseInt (entrada.readLine());
7      y = Integer.parseInt (entrada.readLine());
8      z = Integer.parseInt (entrada.readLine());
9
10     if (x>y && x>z)
11         max = x;
12     else{
13         if (z>y)
14             max = z;
15         else
16             max = y;
17     }
18     System.out.println ("El máximo es "+ max);
19 } //main
20 }
```



Muestra El Número de Mayor Valor.

Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$ , Hay 9 Aristas y 7 Nodos.

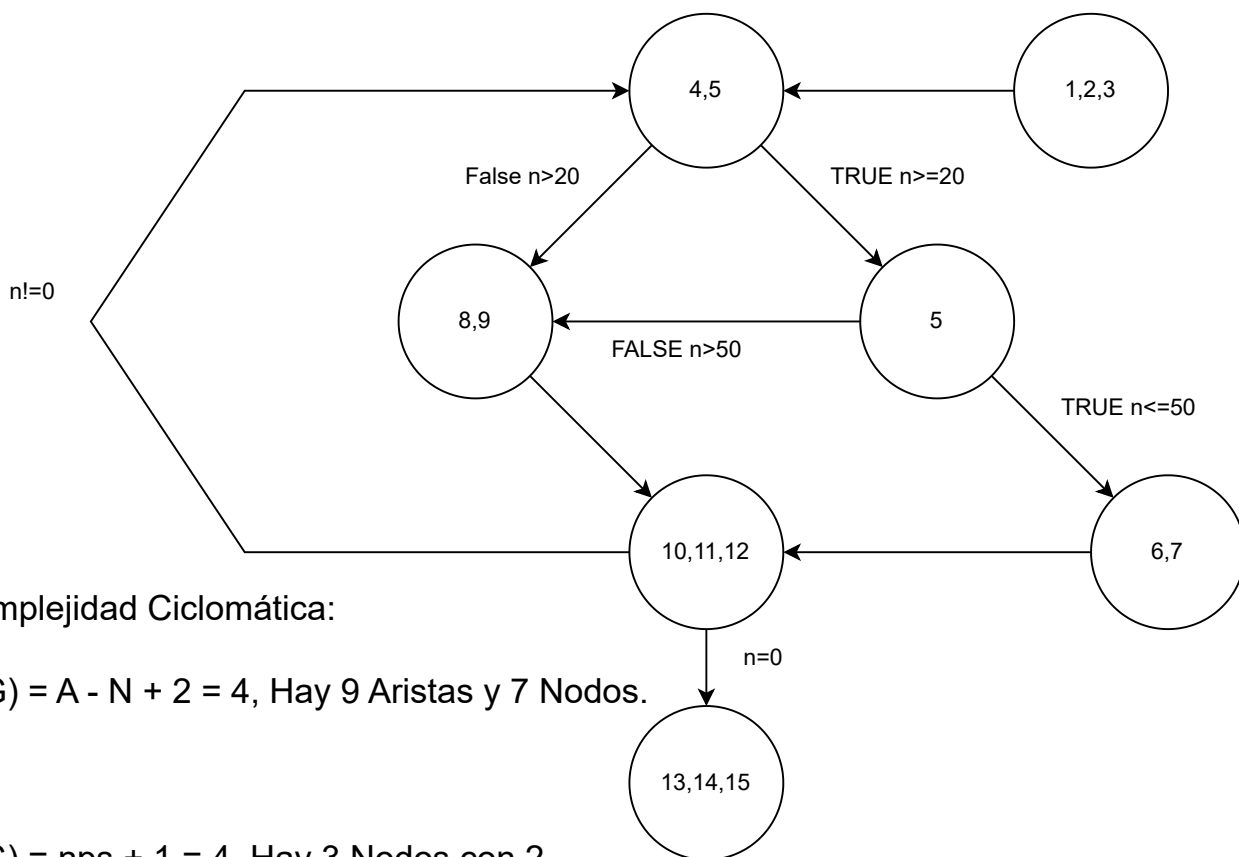
ó

$V(G) = nps + 1 = 4$ , Hay 3 Nodos con 2 Aristas.

Programa Simple sin Mucho Riesgo.

## Ejercicio - 4

```
function obtener_media : real ;  
  
var  
1   n, suma, conta, suma2, total_num : integer ;  
  
2 begin  
3   read( n ) ;  
4   repeat  
5     if (n >= 20 and n <= 50) then  
6       suma := suma + n ;  
7       conta := conta + 1  
8     else  
9       suma2 := suma2 + n ;  
10      total_num := total_num + 1 ;  
11      read (n) ;  
12    until n = 0 ;  
13    obtener_media := suma / conta ;  
14    write (total_num, suma2) ;  
15  end ;
```



Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$ , Hay 9 Aristas y 7 Nodos.

ó

$V(G) = nps + 1 = 4$ , Hay 3 Nodos con 2 Aristas.

Programa Simple sin Mucho Riesgo.

## Ejercicio - 5

Complejidad Ciclomática:

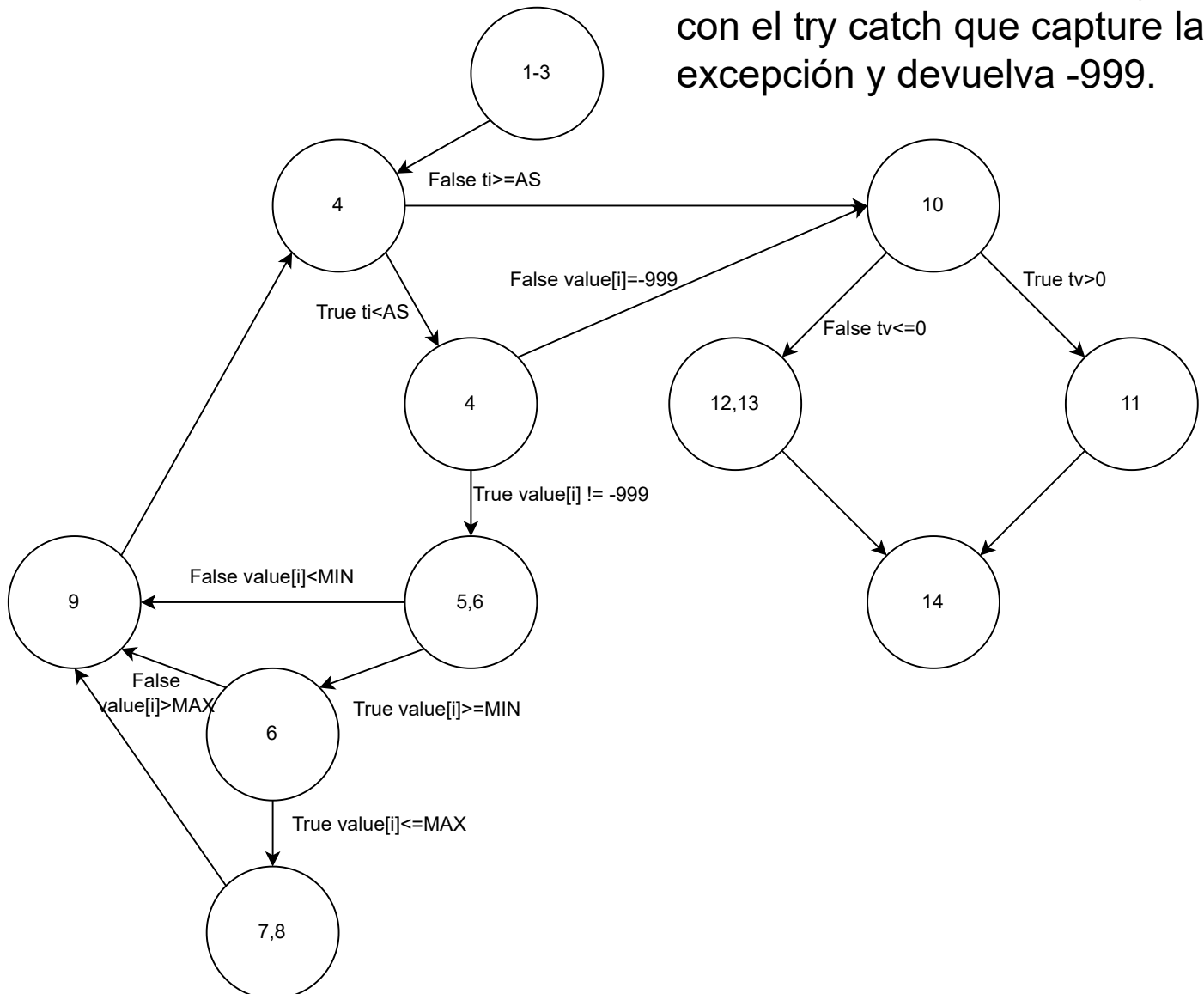
$V(G) = A - N + 2 = 6$  Hay 15 Aristas y 11 Nodos.

$V(G) = nps + 1 = 6$  Hay 5 Nodos con 2 Aristas.

Error de Lógica:

El Código tiene un Error de Lógica, ya que si se le pasa un tamaño de Array mayor al tamaño real, cuando busque en una posición que no existe, dará un excepción de Index Out of Bounds del Array. Falta Implementar el Código con el try catch que capture la excepción y devuelva -999.

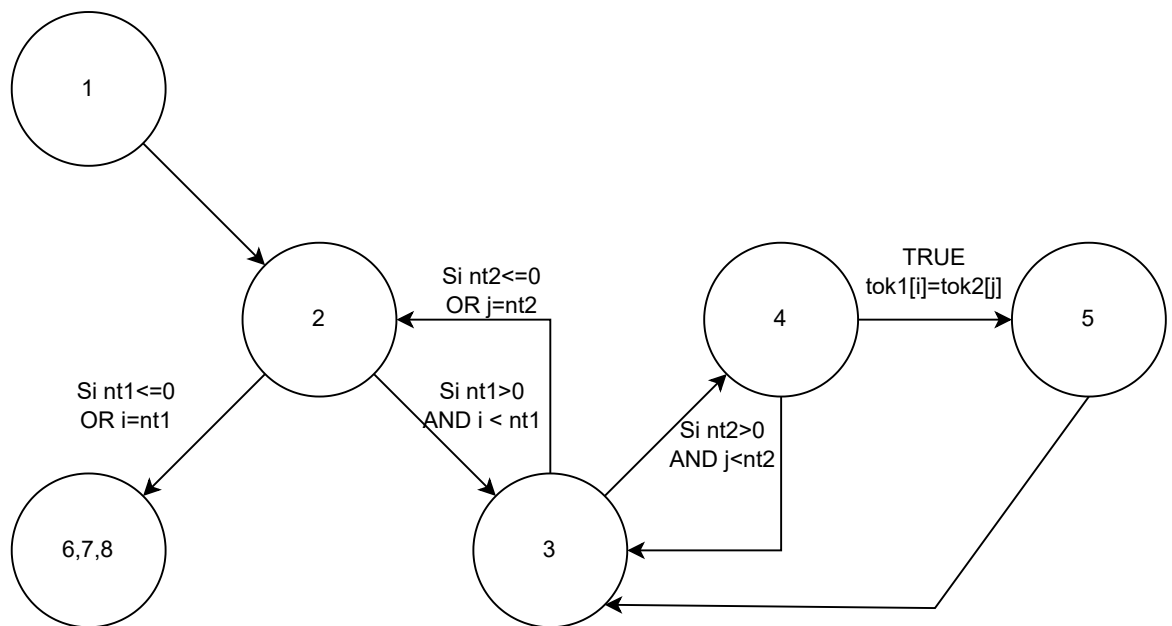
```
public static double ReturnAverage(int value[],
                                   int AS, int MIN, int MAX){
    /*
    Function: ReturnAverage Computes the average
    of all those numbers in the input array in
    the positive range [MIN, MAX]. The maximum
    size of the array is AS. But, the array size
    could be smaller than AS in which case the end
    of input is represented by -999.
    */
    int i, ti, tv, sum;
    double av;
    i = 0; ti = 0; tv = 0; sum = 0;
    while (ti < AS && value[i] != -999) {
        ti++;
        if (value[i] >= MIN && value[i] <= MAX) {
            tv++;
            sum = sum + value[i];
        }
        i++;
    }
    if (tv > 0)
        av = (double)sum/tv;
    else
        av = (double) -999;
    return (av);
}
```



## Ejercicio - 6

Calcula cohesión (int nt1, nt2; String tok1[ ], tok2[ ])

```
1 numAdh = 0
2 Para i de 0 hasta nt1-1
3   Para j de 0 hasta nt2-1
4     Si tok1[i]=tok2[j] entonces
5       numAdh = numAdh + 1
6 total = tok1 + tok2 - numAdh
7 cohesión = numAdh / total
8 regresa cohesión
```



Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$  Hay 8 Aristas y 6 Nodos.

ó

$V(G) = nps + 1 = 4$  Hay 3 Nodos con 2 Aristas.

Programa Simple sin Mucho Riesgo

## Ejercicio -7

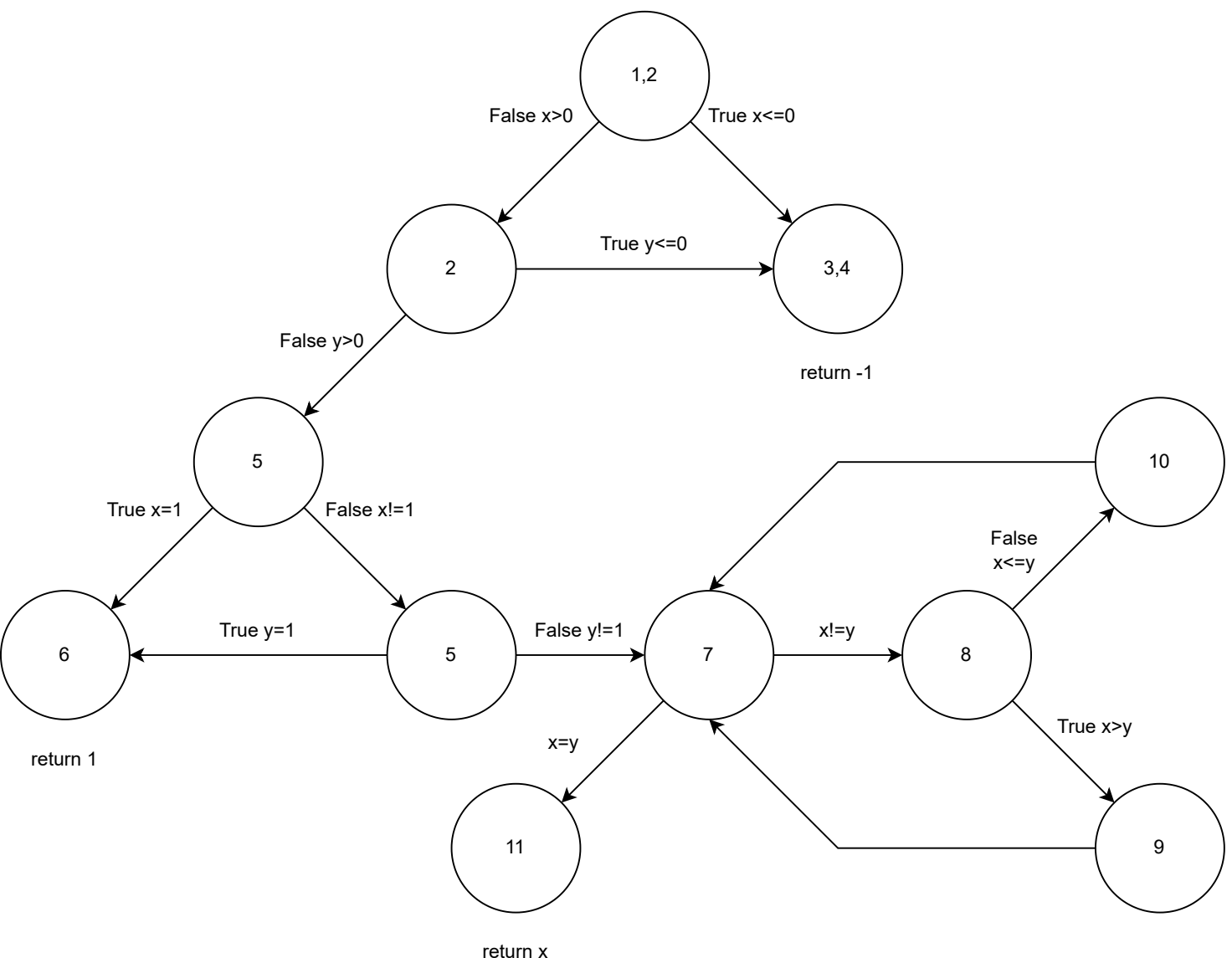
```
1 lee x, y
2 Si (x<=0) o (y<=0) entonces
3   Escribe "deben ser no negativos;
4   regresa -1
5 Si (x=1) o (y=1) entonces
6   regresa 1
7 Mientras (x <> y)
8   Si (x>y) entonces
9     x = x - y
10  de otro modo y = y - x
11 regresa x
```

### Complejidad Ciclomática:

$V(G) = A - N + 2 = 5$  Hay 14 Aristas 11 Nodos

ó

$V(g) = nps + 1 = 5$  Hay 5 Nodos con 2 Aristas, si bien la salida del while muestra 2 Aristas, al ser el único camino no se Considera un Predicado.





## Ejercicio - 8

```
1. Program triangle2 'Structured programming version of simpler specification
2. Dim a,b,c As Integer
3. Dim IsATriangle As Boolean
   'Step 1: Get Input
4. Output("Enter 3 integers which are sides of a triangle")
5. Input(a,b,c)
6. Output("Side A is ",a)
7. Output("Side B is ",b)
8. Output("Side C is ",c)
   'Step 2: Is A Triangle?
9. If (a < b + c) AND (b < a + c) AND (c < a + b)
10. Then IsATriangle = True
11. Else IsATriangle = False
12 EndIf
   'Step 3: Determine Triangle Type
13. If IsATriangle
14. Then If (a = b) AND (b = c)
15. Then Output ("Equilateral")
16. Else If (a ≠ b) AND (a ≠ c) AND (b ≠ c)
17. Then Output ("Scalene")
18. Else Output ("Isosceles")
19. EndIf
20. EndIf
21. Else Output("Not a Triangle")
22 EndIf
23. End triangle2
```

## Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

