

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
```

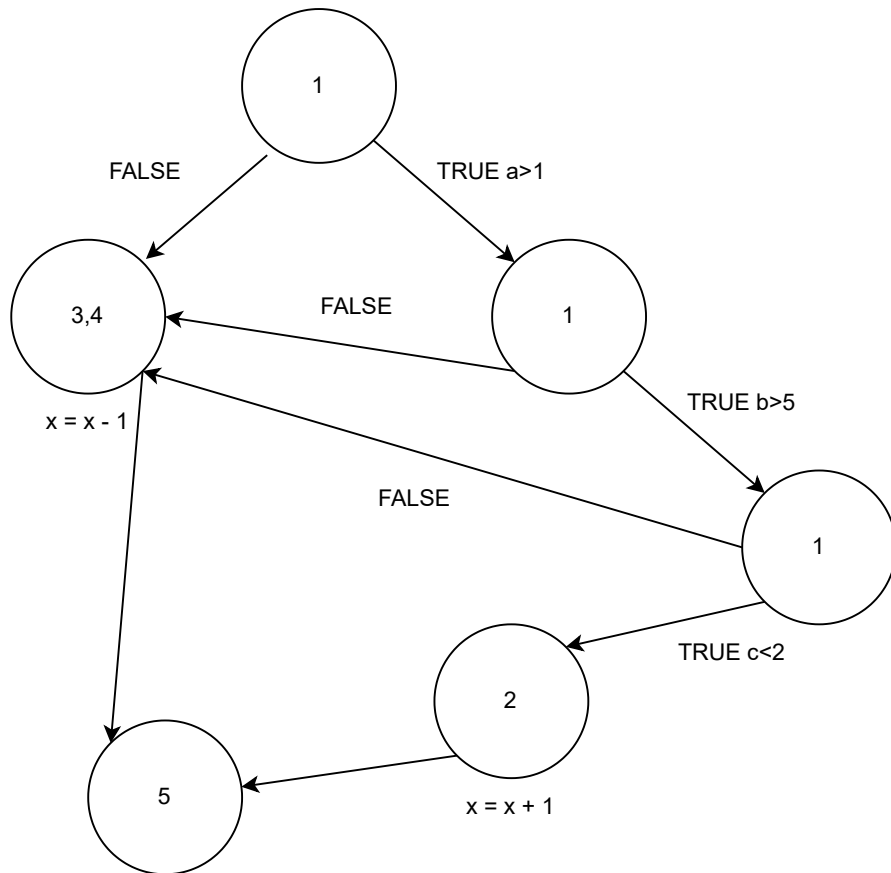
Complejidad Ciclomática:

$V(G) = A - N + 2 = 4$, 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
```

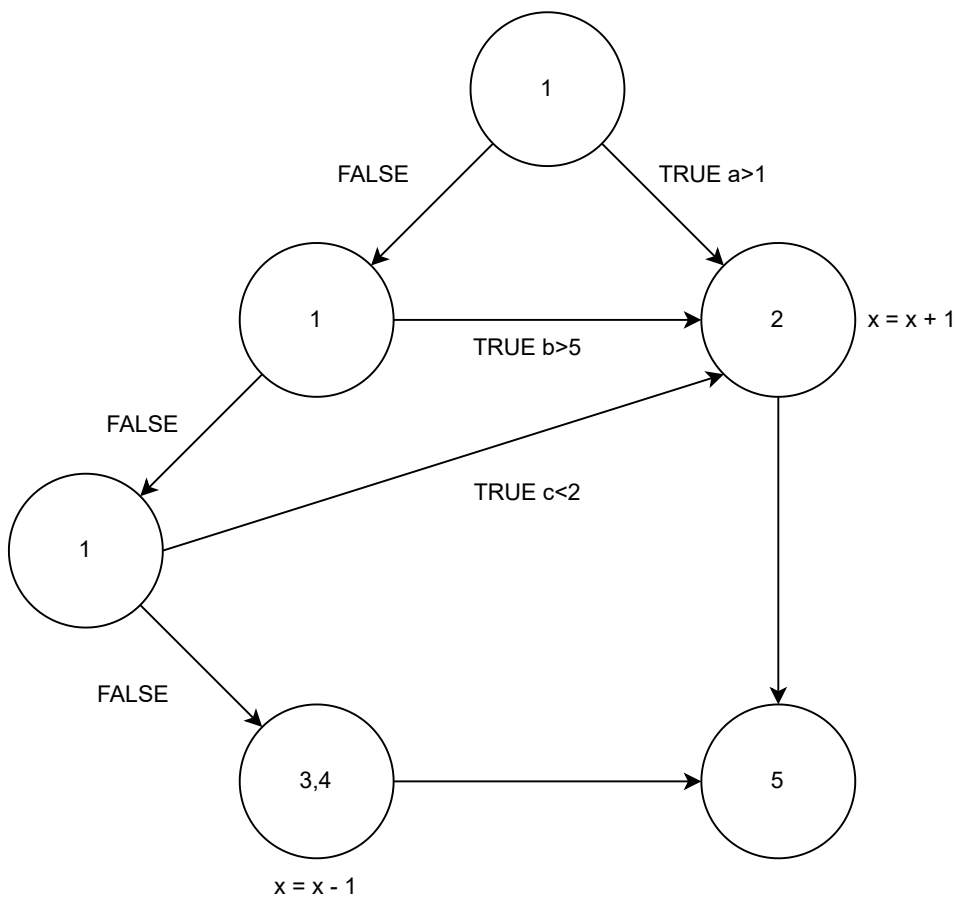
Complejidad Ciclomática:

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

ó

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

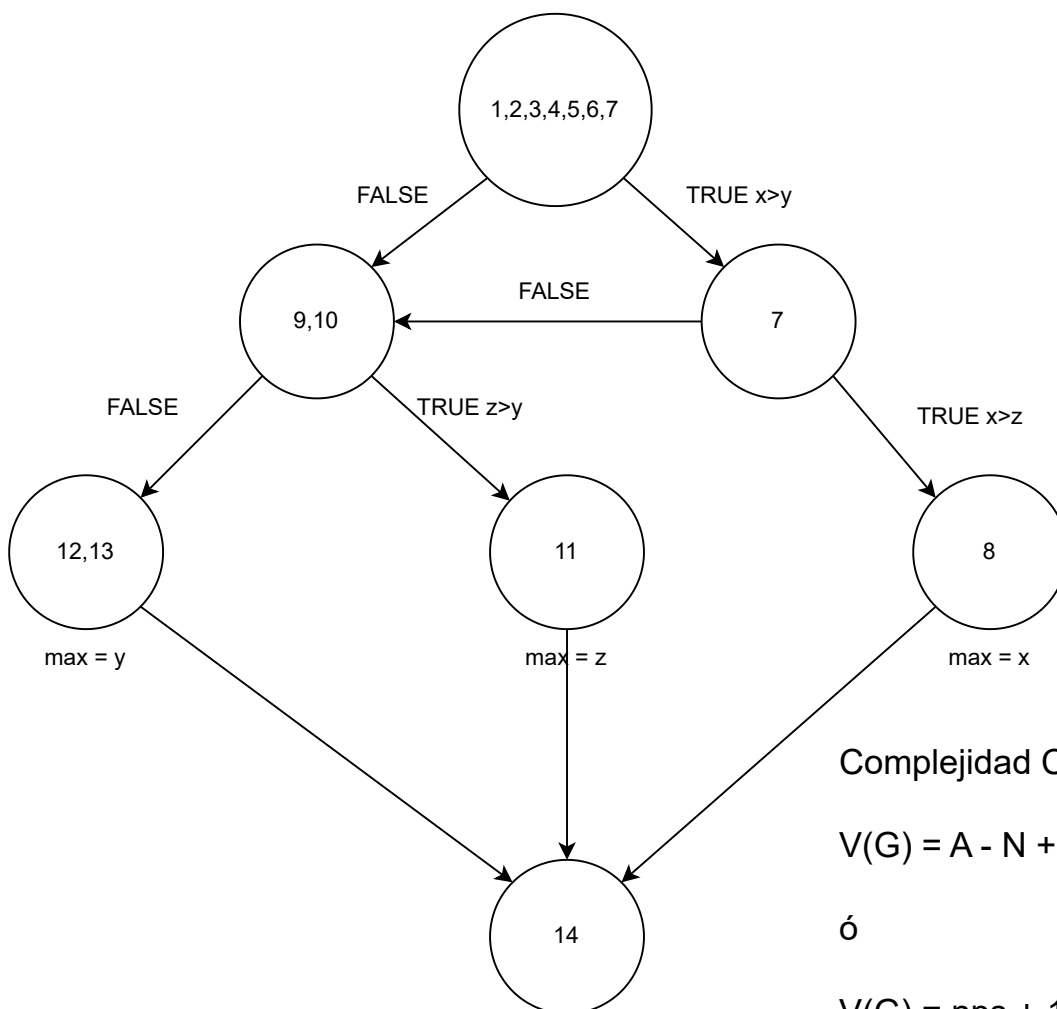
Programa Simple sin Mucho Riesgo.



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 }
```



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.

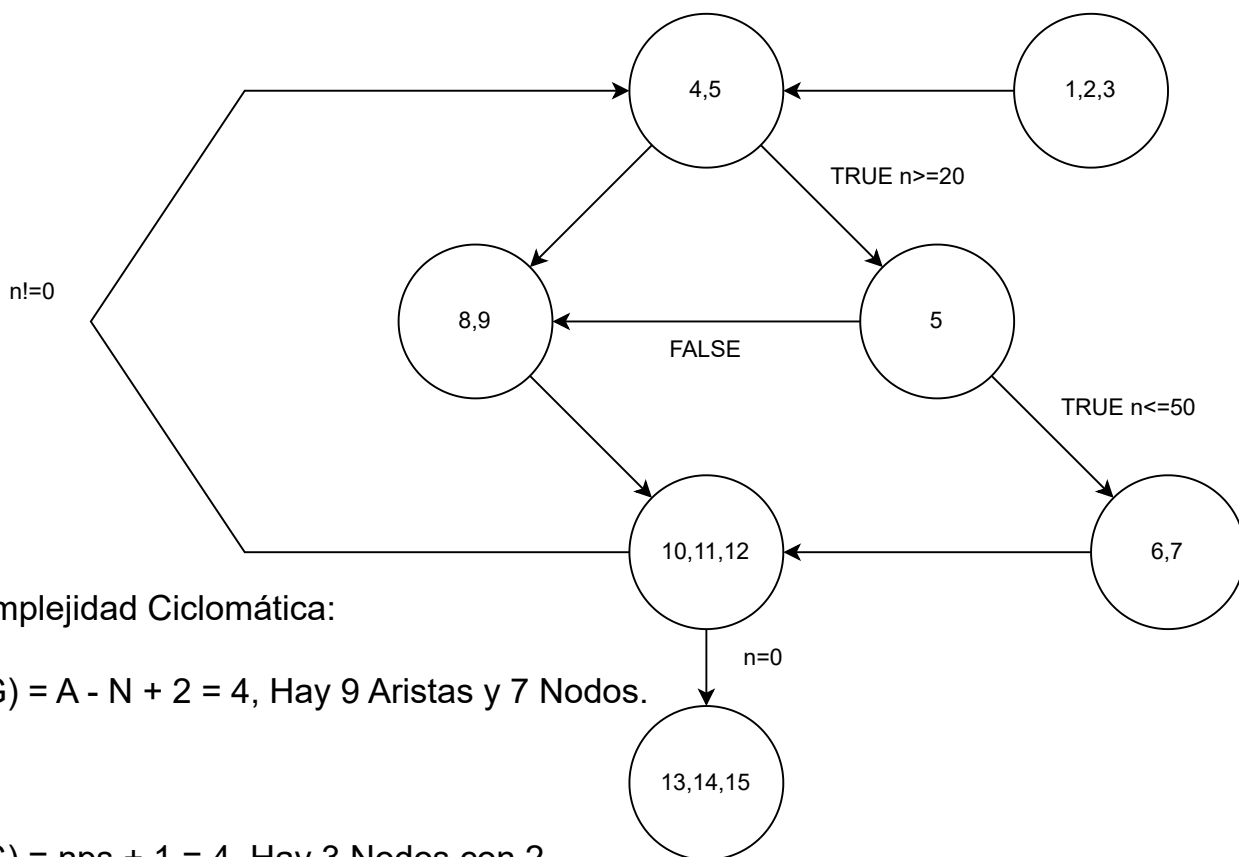
Programa Simple sin Mucho Riesgo.

Ejercicio - 4

```

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

```



Complejidad Ciclomática:

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

ó

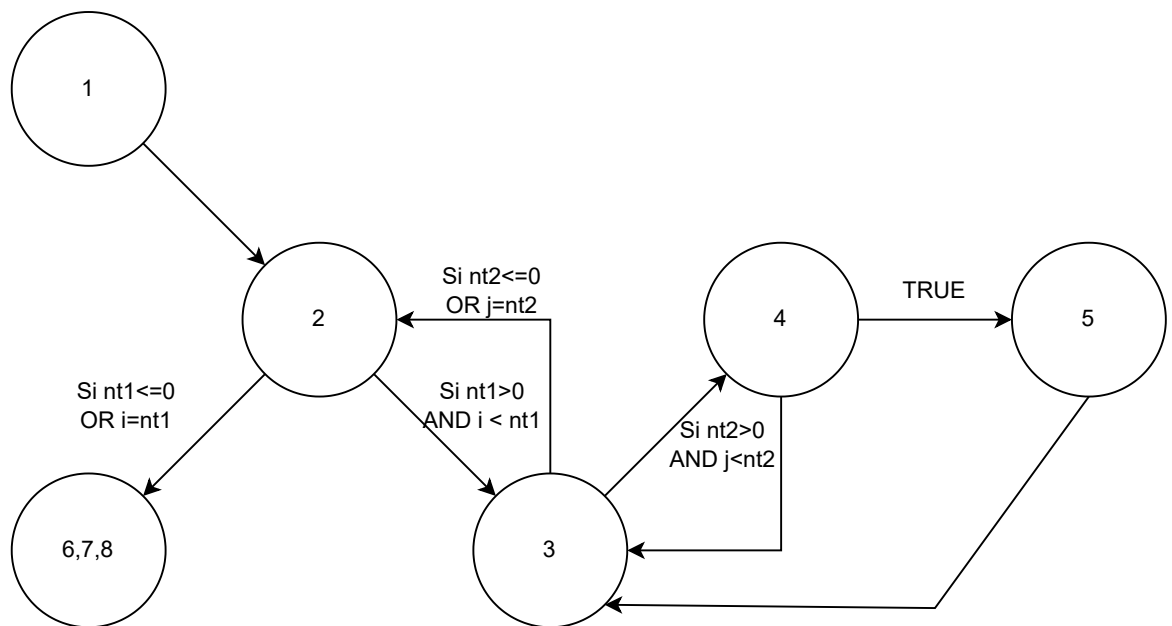
$V(G) = nps + 1 = 4$, Hay 3 Nodos con 2 Aristas.

Programa Simple sin Mucho Riesgo.

Ejercicio - 5

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.