

Indicar si el siguiente multiprograma termina.
Suponga atomicidad línea a línea.
Inicialmente $x, y = 25, 25$.

<pre>while (0 < x + y < 100) { x = x + 1 y = y + 1 }</pre>	<pre>while (0 < x + y < 100) { x = x - 1 y = y + 1 }</pre>	<pre>while (0 < x + y < 100) { x = x + 1 y = y - 1 }</pre>	<pre>while (0 < x + y < 100) { x = x - 1 y = y - 1 }</pre>
--	--	--	--

Para el siguiente multiprograma de dos componentes, **suponga atomicidad línea a línea**.
Inicialmente $x = 4$.

$x = x + 2$	$x = x - 1$
-------------	-------------

Indique que valores finales puede tomar x .

Seleccione una o más de una:

El siguiente es multiprograma de dos componentes.

No suponga atomicidad, es decir cada incremento o decremento es: leer la memoria, operar, escribir en la memoria.
Inicialmente $x = 4$.

$x = x + 2$	$x = x - 1$
-------------	-------------

Indique que valores finales puede tomar x .

Suponga atomicidad línea a línea. La variable i y el arreglo a son compartidos.

Inicialmente $i = 2$ y $a = [2, 2, 2, 2]$.

<pre>while (0 <= i < 4) { a[i] = 0 i++ }</pre>	<pre>while (0 <= i < 4) { a[i] = 1 i-- }</pre>
--	--

Indique si el multiprograma termina.

Suponga atomicidad línea a línea. La variable **i** y el arreglo **a** son compartidos. Inicialmente $i=2$ y $a=[2,2,2,2]$.

```
while(0<=i<4) {      while(0<=i<4) {  
    a[i]=0            a[i]=1  
    i--              i++  
}                    }
```

Indique que valores puede tomar el arreglo **a** del multiprograma cuando termina

Suponga atomicidad línea a línea. La variable **i** y el arreglo **a** son compartidos. Inicialmente $i=2$, $a=[2,2,2,2]$, $s=1$ y $t=0$

```
while(0<=i<4) {      while(0<=i<4) {  
    sem_wait(s)        sem_wait(t)  
    a[i]=0             a[i]=1  
    i--               i++  
    sem_post(t)        sem_post(s)  
}                    }
```

Indique que valores puede tomar el arreglo **a** del multiprograma cuando termina