# Programación Concurrente

## Práctica 1: Trazas

[Recuerde que, salvo que se indique lo contrario, sólo se consideran atómicas la lectura y escritura]

1. Asumiendo que cada instrucción **print** es atómica. Mostrar todas las posibles trazas de ejecuciones del siguiente programa

```
thread T1: {          thread T2: {
  print("Hola");        print("Hola");
  print("Pepe");        print("Juan");
}                     }
```

2. Cuáles son los valores que puede tomar $x$ al final de la ejecución de los siguientes programas.

   *a)*

```
global int x = 0;

thread T1: {              thread T2: {
  int local = x;            int local = x;
  local = local + 1;        local = local + 1;
  x = local;                x = local;
}                         }
```

   *b)*

```
global int x = 0;

thread T1:              thread T2:
  x = x + 1;              x = x + 1;
```

3. Dado el siguiente programa

```
global int x = 0;
global int y = 0;

thread T1:              thread T2:
  y = x + 1;              x = y + 1;
```

   *a)* Mostrar una traza de ejecución en el que los valores de las variables globales al final de la ejecución del programa son $x = 2$ e $y = 1$.

   *b)* Decir si existe una traza de ejecución tal que al final $x = y = 1$. Justificar.

4. Dado el siguiente programa

```
global int n = 0;

thread T1: {            thread T2: {
  int local;              int local;
  repeat (5) {            repeat (5) {
    local = n;              local = n;
    n = local + 1;          n = local + 1;
} }                     } }
```

   - Mostrar una traza de ejecución del programa en el que el valor final de $n$ sea 5.

5. Asumir que la función $f$ tiene una raíz entera, es decir, $f(x) = 0$ para algún valor $x$ entero. A continuación proponemos distintos programas para encontrar tal raíz. Consideraremos a un programa correcto si ambos threads terminan cuando uno de ellos ha encontrado la raíz. Para cada programa, decir si es correcto o no, justificando la respuesta.

   Programa A:

```
global boolean found;

thread T1: {                  thread T2: {
  int i = 0;                    int j = 1;
  found = false;                found = false;
  while (!found) {              while (!found) {
    i = i + 1;                    j = j - 1;
    found = (f(i) == 0);          found = (f(j) == 0);
} }                           } }
```

Programa B:

```
global boolean found = false;

thread T1: {                    thread T2: {
  int i = 0;                      int j = 1;
  while (!found) {                while (!found) {
    i = i + 1;                      j = j - 1;
    found = (f(i) == 0);            found = (f(j) == 0)
} }                             } }
```

Programa C:

```
global boolean found = false;

thread T1: {              thread T2: {
  int i = 0;                int j = 1;
  while (!found) {          while (!found) {
    i = i + 1;                j = j - 1;
    if (f(i) == 0)            if (f(j) == 0)
      found = true;             found = true;
} }                       } }
```

6. Considerar el siguiente programa

```
global int n = 0;

thread T1: {          thread T2: {
  while (n < 2)         n = n + 1;
    print(n);           n = n + 1;
}                     }
```

   *a*) Dar las trazas de ejecución que muestren por pantalla las siguientes secuencias: 012, 002, 02.

   *b*) Debe necesariamente aparecer el valor 2 en la salida?

   *c*) Cuántas veces puede aparecer 2 en la salida?

   *d*) Cuántas veces puede aparecer 1 en la salida?

   *e*) Cuántas veces puede aparecer 0 en la salida?

   *f*) Cual es la longitud de la secuencia mas corta que puede ser mostrada?

7. Considerar el siguiente programa

```
global int n = 0;

thread T1:            thread T2:
  while (n < 1)         while (n >= 0)
    n = n + 1;            n = n - 1;
```

   *a*) Dar una traza en la que el loop en el thread de la izquierda ejecute exactamente una vez.

   *b*) Dar una traza en la que el loop en el thread de la izquierda ejecute exactamente tres vez.

   *c*) Describir una traza en la que el loop en el thread de la izquierda no termine nunca.

8. Considerar el siguiente programa

```
global int n = 0;
global boolean flag = false;

thread T1:            thread T2:
  while (!flag)         while (!flag)
    n = 1 - n;            if (n == 0)
                            flag = true;
```

   *a*) Dar una traza de ejecución en la que el programa termine.

   *b*) Cuáles son los posibles valores de $n$ cuando el programa termina.

   *c*) ¿Puede una ejecución del programa no terminar?