1.

global Semaphore permisoA = new Semaphore (0);

global Semaphore permisoF = new Semaphore (0);

thread {

print ( ’A ’ );

permisoA.release();

print ( ’B ’ );

permisoF.acquire();

print ( ’C ’ );

}

thread {

print ( ’E ’ );

permisoA.acquire();

print ( ’F ’ );

permisoF.release();

print ( ’G ’ );

}

2.

global Semaphore permisoA = new Semaphore (0);

global Semaphore permisoC = new Semaphore (0);

global Semaphore permisoO = new Semaphore (0);

thread {

permisoA.acquire();

print ( ’C ’ );

permisoC.release();

print ( ’E ’ );

permisoO.release();

}

thread {

print ( ’A ’ );

permisoA.release();

permisoC.acquire();

print ( ’R ’ );

permisoO.acquire();

print ( ’O ’ );

}

3.

global Semaphore permisoR = new Semaphore (0);

global Semaphore permisoI = new Semaphore (0);

global Semaphore permisoO = new Semaphore (0);

thread {

print ( " R " );

permisoR.release(2);

permisoI .acquire();

permisoO.acquire();

print ( " OK " );

}

thread {

permisoR.acquire();

print ( " I " );

permisoI .release(2);

permisoO.acquire();

print ( " OK " );

}

thread {

permisoR.acquire();

permisoI .acquire();

print ( " O " );

permisoO .release(2);

print ( " OK " );

}

4.

global Semaphore permisoF = new Semaphore (0);

global Semaphore permisoH = new Semaphore (0);

global Semaphore permisoC = new Semaphore (0);

thread

while (true) {

print ( ’A ’ );

permisoF .release();

print ( ’B ’ );

permisoC .acquire();

print ( ’C ’ );

print ( ’D ’ );

}

thread

while (true) {

print ( ’E ’ );

permisoH .release();

permisoF .acquire();

print ( ’F ’ );

print ( ’G ’ );

permisoC .release();

}

thread

while (true) {

permisoH .acquire();

print ( ’H ’ );

print ( ’I ’ );

}

F <= A

H <= E

C <= G

5.

global Semaphore permisoT1 = new Semaphore (0);

global Semaphore permisoT2 = new Semaphore (0);

global Semaphore permisoFinal = new Semaphore (1);

global int x = 0;

thread T1 : {

permisoFinal .acquire();

x = x + 1;

permisoT1.release(2);

}

thread T2 : {

permisoT1.acquire();

x = x + 2;

permisoT2.release();

}

thread T3 : {

permisoT1.acquire();

permisoT2.acquire();

x = x + 3;

}

x = 6.

6.

global int y = 0 , z = 0;

thread {

int x ;

x = y + z ;

}

thread {

y = 1;

z = 2;

}

a.

X puede tener el valor de 0, 1, 3.

b.

Para x = 0

global Semaphore permiso = new Semaphore (0);

thread {

int x ;

x = y + z ;

permiso.release();

}

thread {

permiso.acquire();

y = 1;

z = 2;

}

Para x = 1

global Semaphore permisoY = new Semaphore (0);

global Semaphore permisoZ = new Semaphore (0);

thread {

int x ;

permisoY.acquire();

x = y + z ;

permisoZ.release();

}

thread {

y = 1;

permisoY.release();

permisoZ.acquire();

z = 2;

}

Para x = 3

global Semaphore permisoY = new Semaphore (0);

global Semaphore permisoZ = new Semaphore (0);

thread {

int x ;

permiso.acquire();

x = y + z ;

permisoZ.release();

}

thread {

y = 1;

z = 2;

permiso.release();

}

7.

a y b.

global Semaphore permisoA = new Semaphore (1);

global Semaphore permisoB = new Semaphore (0);

thread

while (true)

permisoA.acquire();

print ( ’A ’ );

permisoB.release();

thread

while (true)

permisoB.acquire();

print ( ’B ’ );

permisoA.release();

8.

global int N = 50 , N2 = 0;

global Semaphore permisoResta = new Semaphore (0);

global Semaphore permisoSuma = new Semaphore (1);

global Semaphore permisoPrint = new Semaphore (1);

thread {

while ( N > 0)

permisoResta.acquire();

N = N -1;

permisoSuma.release();

permisoResta.acquire();

print ( N2 );

}

thread

while (true)

permisoSuma.acquire();

N2 = N2 + 2\* N + 1;

permisoResta.release();