**1- Quelle est la méthode qui permet de démarrer un thread :** start() ;   
 run() ;

**2- Pour assurer la synchronisation du point de rendez-vous suivant, déterminer les solutions correctes Pour Pt de rdv (init(SA,0), init (SB,0)):**

|  |  |
| --- | --- |
| **Processus A** | **Processus B** |
| **I1; P(SB); V(SA); I2;** | **J1; P(SA); V(SB); J2;** |

**3- Qu’implémente la classe Semaphore :**Object ; **4- Parmi les propositions suivantes déterminer celles qui sont correctes :**  
♦ Les méthodes synchronized sont remplacées par les méthodes lock(), unlock() etc. de l’interface Lock ;  
♦ Dans une classe ayant une partie du code synchronisé, plusieurs threads peuvent accéder à l’autre partie du code non synchronisé ;  
♦ Pour une barrière utilisée une seule fois, il vaut mieux utiliser CountDownLatch ;  
♦ Les variables à conditions sont gérées par le paquetage java.util.concurrent.lock ;  
♦ Une barrière permet de bloquer plusieurs thread en un point jusqu’à ce qu’un nombre prédéfini de threads atteigne ce point ;  
♦ Quand le thread est endormi, il libère ses verrous ;

**5- Que va afficher ce code ?  
  
class Test extends Thread{  
 public void run(){  
 for(int i = 0; i < 30; i++)  
 System.out.println("LMD GL");  
 }  
 public static void main(String[] args){  
 Test t = new Test();  
 }  
}** Rien du tout.

**6- Quand un thread est-il considéré comme mort ?**c. Lorsqu'il a dépilé la méthode run() de sa pile d'exécution.

**Exercice 2**

/\* Sémaphores et initialisation  
 semaphore ville1=1

Semaphore ville2=1

Semaphore v2=1

Semaphore v1=1

Int v1v2=0

Int v2v1=0  
Demande d’accès à la voie par la VILLE1 () {  
 V1v2++

If(v1v2 !=0){

p(v1)

}

If(v1v2 !=0){

p(v1)

}

}  
Sortie de la voie par la VILLE 2 (){.  
 v1v2—

If(v1v2 !=0){

v(v2)

v(v1)

}

}  
Demande d’accès à la voie par la VILLE2 () {

V2v1++

If(v2v1 !=0){

p(v2)

}

If(v1v2 !=0){

p(v2)

}

}  
Sortie de la voie par la VILLE1 (){

V2v1—

If(v1v2 !=0){

v(v1)

v(v2)

}  
}

**Exercice 3**

Void Produire(m){

if(cp == N){

wait(nplein) ;

}

Tampon[ip]=m ;

Ip=(ip+1)%N ;

Cp++ ;

If(cp==1)

Signale(nvide) ;

}

Void consommer(m){

If (cp==0)

Wait(nvide)

m=tampon[ic];

ic=(ic+1)%N ;

cp-- ;

if (cp==N)

signal(nplein) ;

}