1. Si tous les philosophes prendre au meme temps la fourchette gauche (au droite) donc il y aura un interblocage et le program s’arrete pcq chaque philosophe prendre la forchette droite de l’autre et chaq’un prendre une fourchette et attent la deuxieme fourchette

**Deuxieme solution :**

prendre\_fourchette ( int i ){

if ( i == N-1) { // N-1 : le philosophe gaucher

fourch[ (i + 1) % N ].P( );//prendre la fourchette droite d’abord

fourch[ i ].P( ); //puis il prendre la fourchette gauche

}

else{

fourch[ i ].P( );

fourch[ (i + 1) % N ].P( );

}

}

* Ui poser\_fourchette est la même

**Troisieme solution :**

test\_mange( int i ){

if (Etat\_Philo[i]==A\_FAIM && Etat\_Philo[(i+1)%N] != MANGE && Etat\_Philo[(i-1)%N] != MANGE ) {

Etat\_Philo[i ] =mange;

philo\_prive[ i ].v( );

}

}

prendre\_fourchette (int i){

mutex.P( );

Etat\_Philo[ i ] = A\_FAIM;

test\_mange( i );

mutex.V( );

philo\_prive[ i ].P( );

}

poser\_fourchette (int i){

mutex.P( );

Etat\_Philo[ i ] = PENSE;

test\_mange((i+1)%N );

test\_mange((i-1+N)%N );

mutex.V( );

}

**Quatirieme solution :**

sem s = N-1;

prendre\_fourchette ( int i ){

s.P( );

fourch[ i ].P( );

fourch[ (i + 1) % N ].P( );

s.V( );

}

**Cinqieme solution :**

prendre\_fourchette ( int i ){

while ( Etat\_Philo[ (i+1)%N ] == MANGE || Etat\_Philo[ (i-1+N)%N ] == MANGE ){

file\_privee[ i ].wait( );

}

Etat\_Philo = MANGE;

}

poser\_fourchette ( int i ){

if ( Etat\_Philo[ (i+2)%N ] == PENSE )

file\_privee[ (i+1)%N ].signal( );

}

if ( Etat\_Philo[ (i-2+N)%N ] == PENSE ) {

file\_privee[ (i-1+N)%N ].signal;

}

}