

- 1- Si tous les philosophes prennent au même temps la fourchette gauche (au droite) donc il y aura un interblocage et le programme s'arrête car chaque philosophe prend la fourchette droite de l'autre et chacun prend une fourchette et attend la deuxième 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 prend 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_Philos[i]==A_FAIM && Etat_Philos[(i+1)%N] != MANGE && Etat_Philos[(i-1)%N] != MANGE ) {  
        Etat_Philos[i] =mange;  
        philo_privé[ i ].v( );  
    }  
}
```

```
prendre_fourchette (int i){  
    mutex.P( );  
    Etat_Philos[ i ] = A_FAIM;  
    test_mange( i );  
    mutex.V( );  
    philo_privé[ i ].P( );  
}
```

```
poser_fourchette (int i){  
    mutex.P( );
```

```

    Etat_Philos[i] = PENSE;

    test_mange((i+1)%N);

    test_mange((i-1+N)%N);

    mutex.V();
}

```

### **Quatrieme solution :**

```

sem s = N-1;

prendre_fourchette ( int i ){

    s.P();

    fourch[i].P();

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

    s.V();

}

```

### **Cinquieme solution :**

```

prendre_fourchette ( int i ){

    while ( Etat_Philos[(i+1)%N] == MANGE || Etat_Philos[(i-1+N)%N] == MANGE ){

        file_privée[i].wait();

    }

    Etat_Philos[i] = MANGE;

}

poser_fourchette ( int i ){

    if ( Etat_Philos[(i+2)%N] == PENSE )

        file_privée[(i+1)%N].signal();

    }

    if ( Etat_Philos[(i-2+N)%N] == PENSE ) {

        file_privée[(i-1+N)%N].signal();

    }

}

```