```
while[1]{
     think
     wait(mutex)
          wait(f[i])
          wait(f[i+1%5])
     signal(mutex)
     eat
     signal(f[i])
     signal(f[i+1%5])
}
```

This solution to the dining philosophers problem satisfies all the requirements for a solution. This method only allows for 1 philosopher to pick up forks at a time, until that philosopher has picked up 2 forks. This means that there can only ever be 1 philosopher waiting for a 2nd fork. This means that a deadlock where each philosopher is holding 1 fork and waiting for another to put 1 down will never happen. One problem with this solution is that it is inefficient. If philosopher 2 tries to pick up forks while philosopher 1 is eating, it must wait, then philosopher 3 must wait for philosopher 2 to finish, and so on, until philosopher 5, who could be starved for a while. Basically, the worst case scenario is that there is almost no parallelism.