OS LAB CAT

Name: Kulvir Singh
Register Number: 19BCE2074
Question Number = 74 mod 6 = 2

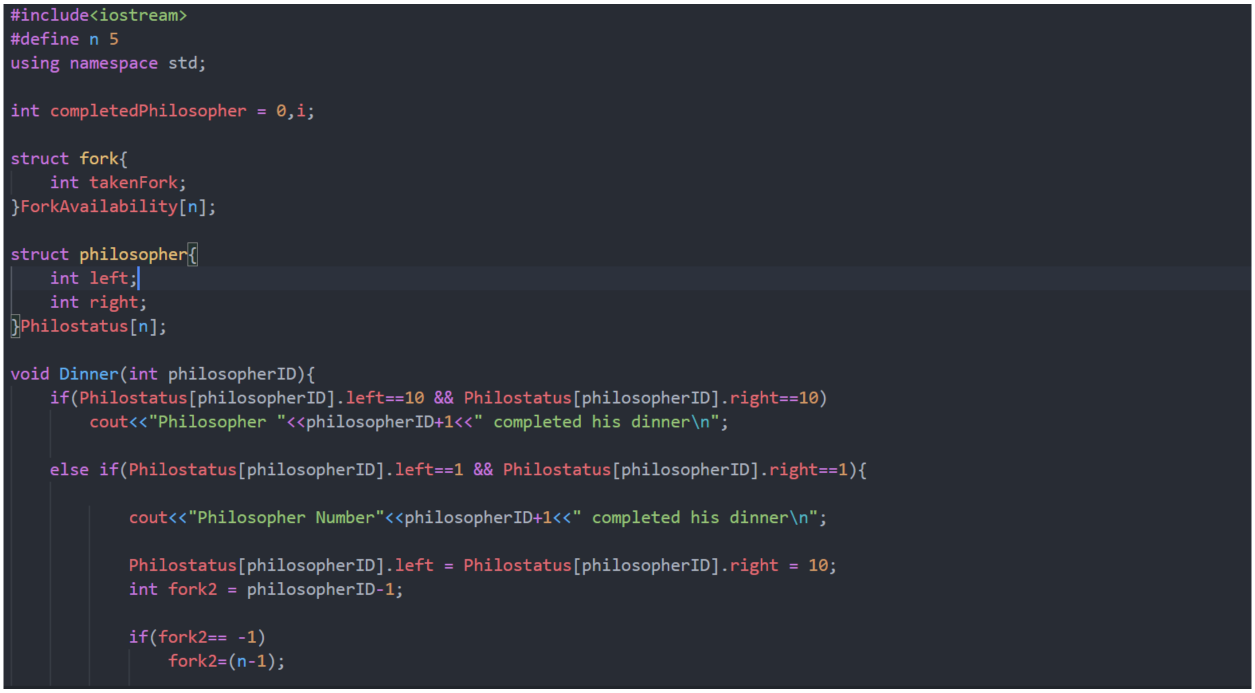
Question 2 : Implement the solution to **Dining Philosophers problem** to illustrate the problem of deadlock or starvation that can occur when many synchronized threads are competing for limited resources.

Solution

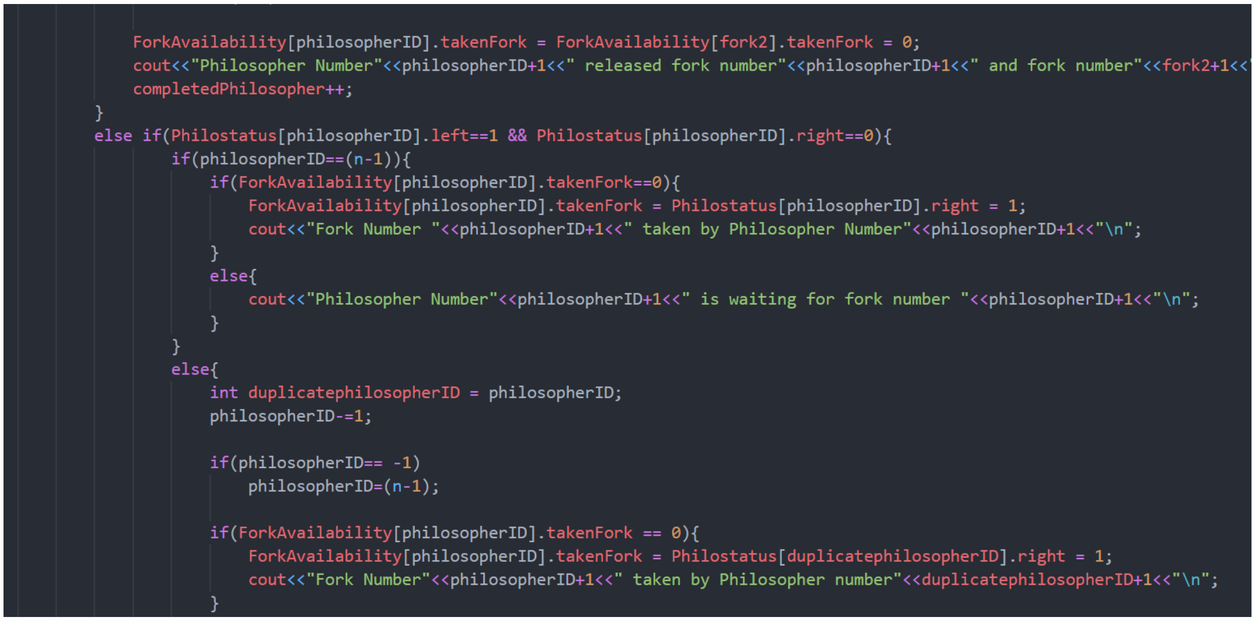
Code:

```
#include<iostream>
#define n 5
using namespace std;

int completedPhilosopher = 0,i;

struct fork{
        int takenFork;
}ForkAvailability[n];

struct philosopher{
        int left;
        int right;
}Philostatus[n];

void Dinner(int philosopherID){
        if(Philostatus[philosopherID].left==10 && Philostatus[philosopherID].right==10)
    cout<<"Philosopher "<<philosopherID+1<<" completed his dinner\n";

        else if(Philostatus[philosopherID].left==1 && Philostatus[philosopherID].right==1){

      cout<<"Philosopher Number"<<philosopherID+1<<" completed his dinner\n";

      Philostatus[philosopherID].left = Philostatus[philosopherID].right = 10;
      int fork2 = philosopherID-1;

      if(fork2== -1)
        fork2=(n-1);
      ForkAvailability[philosopherID].takenFork = ForkAvailability[fork2].takenFork = 0;
      cout<<"Philosopher Number"<<philosopherID+1<<" released fork number"<<philosopherID+1<<"
and fork number"<<fork2+1<<"\n";
      completedPhilosopher++;
    }
    else if(Philostatus[philosopherID].left==1 && Philostatus[philosopherID].right==0){
        if(philosopherID==(n-1)){
          if(ForkAvailability[philosopherID].takenFork==0){
            ForkAvailability[philosopherID].takenFork = Philostatus[philosopherID].right = 1;
            cout<<"Fork Number "<<philosopherID+1<<" taken by Philosopher
Number"<<philosopherID+1<<"\n";
          }
          else{
            cout<<"Philosopher Number"<<philosopherID+1<<" is waiting for fork number
"<<philosopherID+1<<"\n";
          }
        }
        else{
          int duplicatephilosopherID = philosopherID;
          philosopherID-=1;

          if(philosopherID== -1)
            philosopherID=(n-1);

          if(ForkAvailability[philosopherID].takenFork == 0){
            ForkAvailability[philosopherID].takenFork = Philostatus[duplicatephilosopherID].right = 1;
            cout<<"Fork Number"<<philosopherID+1<<" taken by Philosopher
number"<<duplicatephilosopherID+1<<"\n";
          }
          else{
            cout<<"Philosopher number"<<duplicatephilosopherID+1<<" is waiting for Fork
number"<<philosopherID+1<<"\n";
          }
        }
      }
      else if(Philostatus[philosopherID].left==0){
          if(philosopherID==(n-1)){
            if(ForkAvailability[philosopherID-1].takenFork==0){
              ForkAvailability[philosopherID-1].takenFork = Philostatus[philosopherID].left = 1;
              cout<<"Fork number"<<philosopherID<<" taken by philosopher
number"<<philosopherID+1<<"\n";
            }
            else{
              cout<<"Philosopher number"<<philosopherID+1<<" is waiting for fork
number"<<philosopherID<<"\n";
            }
          }
          else{
            if(ForkAvailability[philosopherID].takenFork == 0){
              ForkAvailability[philosopherID].takenFork = Philostatus[philosopherID].left = 1;
              cout<<"Fork number"<<philosopherID+1<<" taken by Philosopher
number"<<philosopherID+1<<"\n";
            }
            else{
              cout<<"Philosopher number "<<philosopherID+1<<" is waiting for Fork number
"<<philosopherID+1<<"\n";
            }
          }
    }
}

int main(){
       for(i=0;i<n;i++)
    ForkAvailability[i].takenFork=Philostatus[i].left=Philostatus[i].right=0;

       while(completedPhilosopher<n){
              for(i=0;i<n;i++)
      Dinner(i);
              cout<<"\nTill now num of philosophers completed dinner are
"<<completedPhilosopher<<"\n\n";
       }
       return 0;
}
```

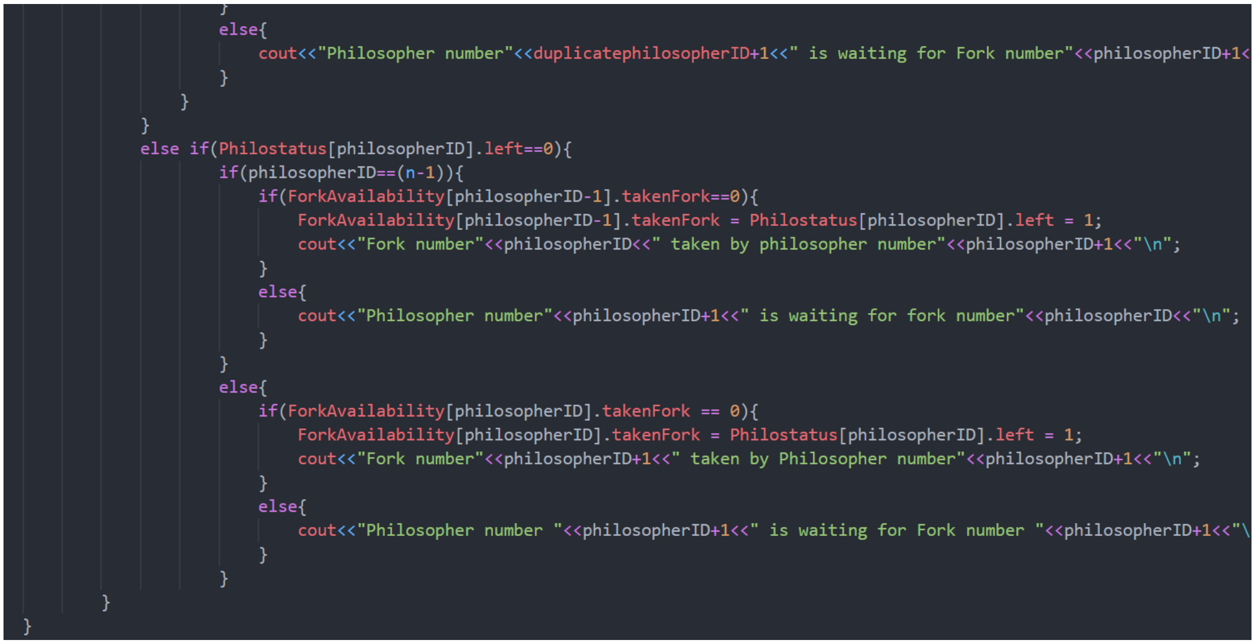
Code Screenshot



```
#include<iostream>
#define n 5
using namespace std;

int completedPhilosopher = 0,i;

struct fork{
    int takenFork;
}ForkAvailability[n];

struct philosopher{
    int left;
    int right;
}Philostatus[n];

void Dinner(int philosopherID){
    if(Philostatus[philosopherID].left==10 && Philostatus[philosopherID].right==10)
        cout<<"Philosopher "<<philosopherID+1<<" completed his dinner\n";

    else if(Philostatus[philosopherID].left==1 && Philostatus[philosopherID].right==1){

            cout<<"Philosopher Number"<<philosopherID+1<<" completed his dinner\n";

            Philostatus[philosopherID].left = Philostatus[philosopherID].right = 10;
            int fork2 = philosopherID-1;

            if(fork2== -1)
                fork2=(n-1);
```

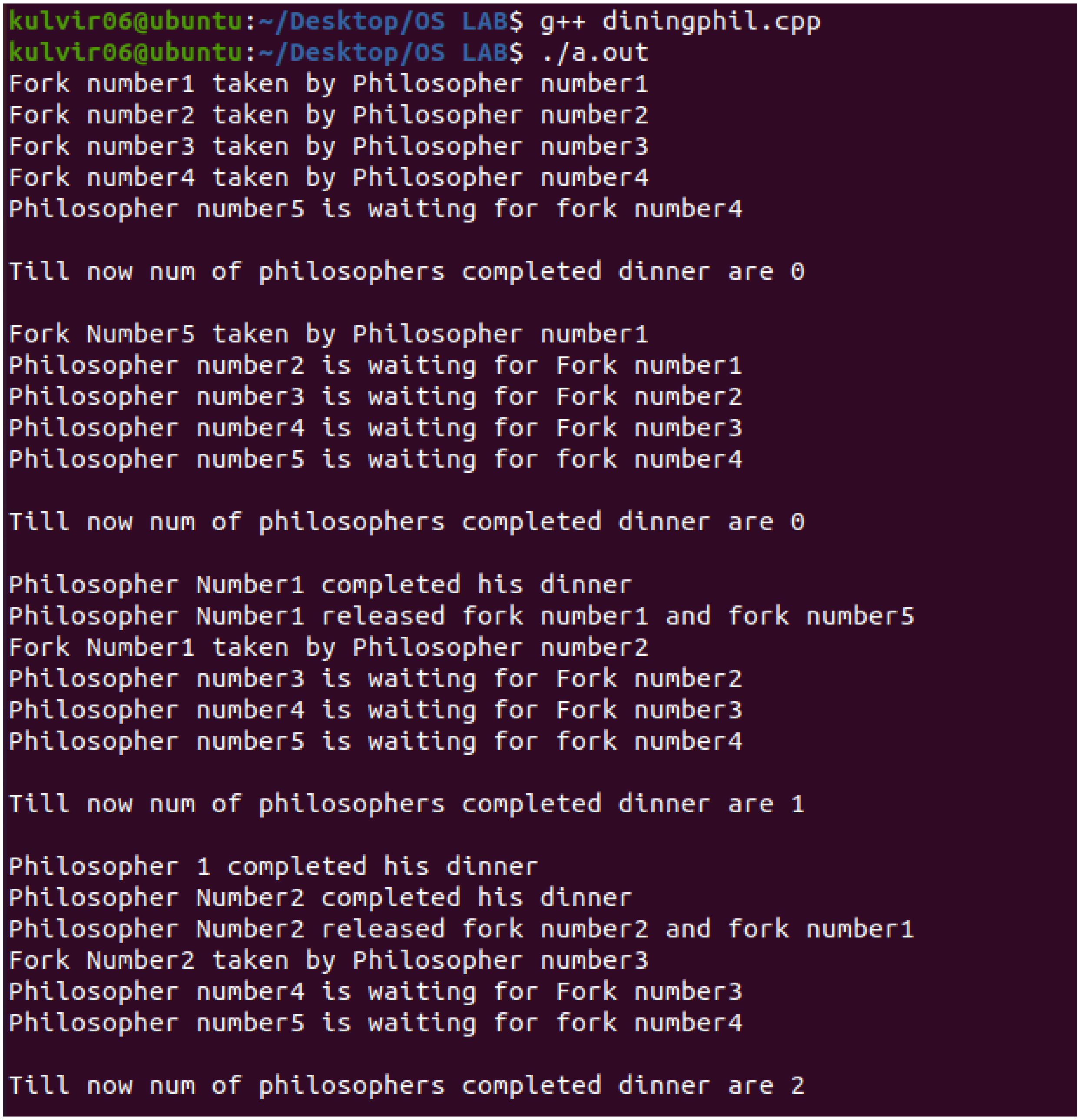


```
            ForkAvailability[philosopherID].takenFork = ForkAvailability[fork2].takenFork = 0;
            cout<<"Philosopher Number"<<philosopherID+1<<" released fork number"<<philosopherID+1<<" and fork number"<<fork2+1<<
            completedPhilosopher++;
        }
        else if(Philostatus[philosopherID].left==1 && Philostatus[philosopherID].right==0){
                if(philosopherID==(n-1)){
                    if(ForkAvailability[philosopherID].takenFork==0){
                        ForkAvailability[philosopherID].takenFork = Philostatus[philosopherID].right = 1;
                        cout<<"Fork Number "<<philosopherID+1<<" taken by Philosopher Number"<<philosopherID+1<<"\n";
                    }
                    else{
                        cout<<"Philosopher Number"<<philosopherID+1<<" is waiting for fork number "<<philosopherID+1<<"\n";
                    }
                }
                else{
                    int duplicatephilosopherID = philosopherID;
                    philosopherID-=1;

                    if(philosopherID== -1)
                        philosopherID=(n-1);

                    if(ForkAvailability[philosopherID].takenFork == 0){
                        ForkAvailability[philosopherID].takenFork = Philostatus[duplicatephilosopherID].right = 1;
                        cout<<"Fork Number"<<philosopherID+1<<" taken by Philosopher number"<<duplicatephilosopherID+1<<"\n";
                    }
```

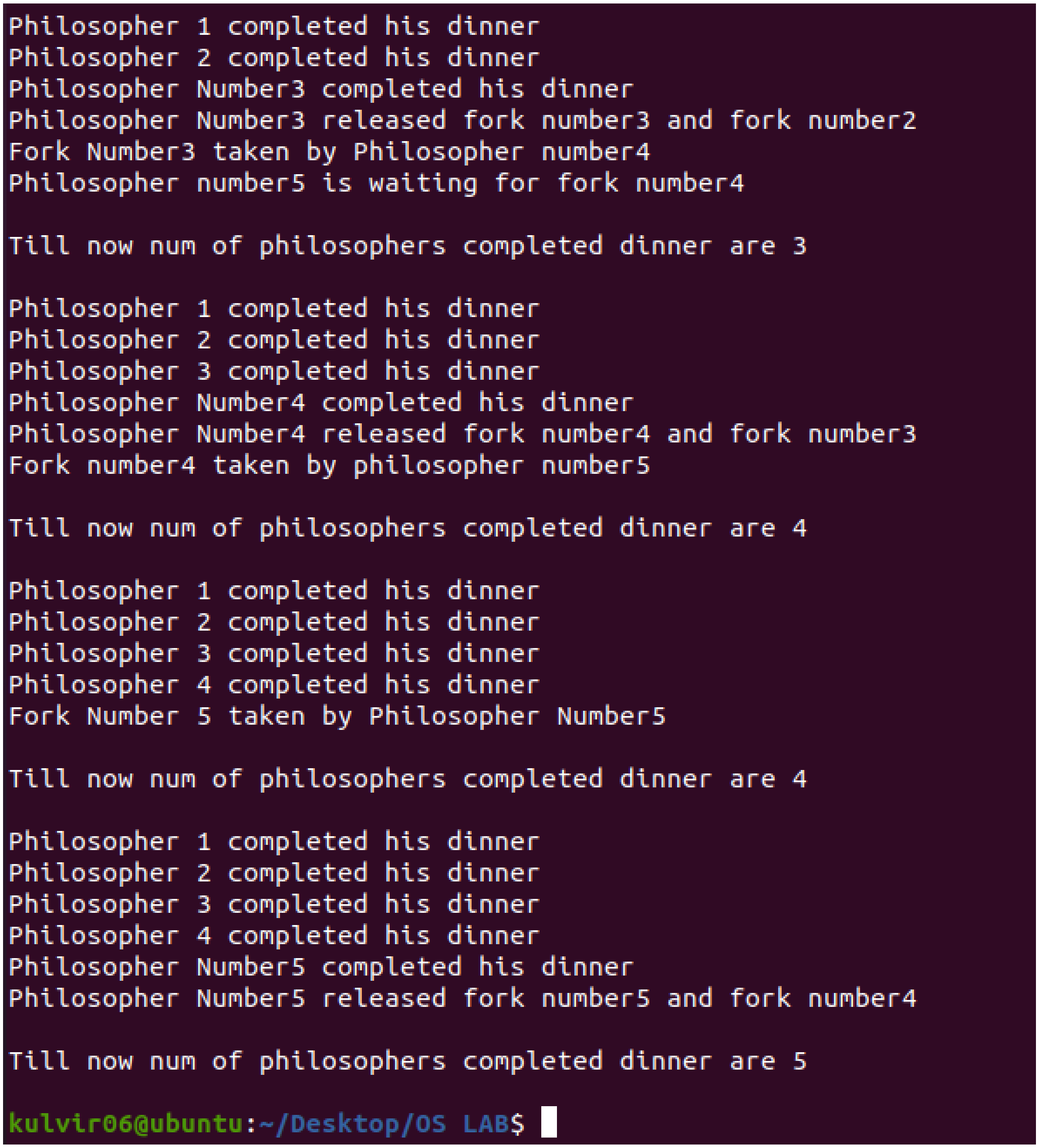


```cpp
                    }
                    else{
                        cout<<"Philosopher number"<<duplicatephilosopherID+1<<" is waiting for Fork number"<<philosopherID+1<
                    }
                }
            }
            else if(Philostatus[philosopherID].left==0){
                    if(philosopherID==(n-1)){
                        if(ForkAvailability[philosopherID-1].takenFork==0){
                            ForkAvailability[philosopherID-1].takenFork = Philostatus[philosopherID].left = 1;
                            cout<<"Fork number"<<philosopherID<<" taken by philosopher number"<<philosopherID+1<<"\n";
                        }
                        else{
                            cout<<"Philosopher number"<<philosopherID+1<<" is waiting for fork number"<<philosopherID<<"\n";
                        }
                    }
                    else{
                        if(ForkAvailability[philosopherID].takenFork == 0){
                            ForkAvailability[philosopherID].takenFork = Philostatus[philosopherID].left = 1;
                            cout<<"Fork number"<<philosopherID+1<<" taken by Philosopher number"<<philosopherID+1<<"\n";
                        }
                        else{
                            cout<<"Philosopher number "<<philosopherID+1<<" is waiting for Fork number "<<philosopherID+1<<"\
                        }
                    }
        }
}
```

Output Screenshot



```
kulvir06@ubuntu:~/Desktop/OS LAB$ g++ diningphil.cpp
kulvir06@ubuntu:~/Desktop/OS LAB$ ./a.out
Fork number1 taken by Philosopher number1
Fork number2 taken by Philosopher number2
Fork number3 taken by Philosopher number3
Fork number4 taken by Philosopher number4
Philosopher number5 is waiting for fork number4

Till now num of philosophers completed dinner are 0

Fork Number5 taken by Philosopher number1
Philosopher number2 is waiting for Fork number1
Philosopher number3 is waiting for Fork number2
Philosopher number4 is waiting for Fork number3
Philosopher number5 is waiting for fork number4

Till now num of philosophers completed dinner are 0

Philosopher Number1 completed his dinner
Philosopher Number1 released fork number1 and fork number5
Fork Number1 taken by Philosopher number2
Philosopher number3 is waiting for Fork number2
Philosopher number4 is waiting for Fork number3
Philosopher number5 is waiting for fork number4

Till now num of philosophers completed dinner are 1

Philosopher 1 completed his dinner
Philosopher Number2 completed his dinner
Philosopher Number2 released fork number2 and fork number1
Fork Number2 taken by Philosopher number3
Philosopher number4 is waiting for Fork number3
Philosopher number5 is waiting for fork number4

Till now num of philosophers completed dinner are 2
```



```
Philosopher 1 completed his dinner
Philosopher 2 completed his dinner
Philosopher Number3 completed his dinner
Philosopher Number3 released fork number3 and fork number2
Fork Number3 taken by Philosopher number4
Philosopher number5 is waiting for fork number4

Till now num of philosophers completed dinner are 3

Philosopher 1 completed his dinner
Philosopher 2 completed his dinner
Philosopher 3 completed his dinner
Philosopher Number4 completed his dinner
Philosopher Number4 released fork number4 and fork number3
Fork number4 taken by philosopher number5

Till now num of philosophers completed dinner are 4

Philosopher 1 completed his dinner
Philosopher 2 completed his dinner
Philosopher 3 completed his dinner
Philosopher 4 completed his dinner
Fork Number 5 taken by Philosopher Number5

Till now num of philosophers completed dinner are 4

Philosopher 1 completed his dinner
Philosopher 2 completed his dinner
Philosopher 3 completed his dinner
Philosopher 4 completed his dinner
Philosopher Number5 completed his dinner
Philosopher Number5 released fork number5 and fork number4

Till now num of philosophers completed dinner are 5

kulvir06@ubuntu:~/Desktop/OS LAB$
```