Dining philospher

#include<iostream>

#define n 4

using namespace std;

int compltedPhilo = 0,i;

struct fork{

int taken;

}ForkAvil[n];

struct philosp{

int left;

int right;

}Philostatus[n];

void goForDinner(int philID){

if(Philostatus[philID].left==10 && Philostatus[philID].right==10){

cout<<"Philosopher "<<philID+1<<" completed his dinner\n";}

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

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

Philostatus[philID].left = Philostatus[philID].right = 10;

int otherFork = philID-1;

if(otherFork== -1)

otherFork=(n-1);

ForkAvil[philID].taken = ForkAvil[otherFork].taken = 0;

cout<<"Philosopher "<<philID+1<<" released fork "<<philID+1<<" and fork "<<otherFork+1<<"\n";

compltedPhilo++;

}

else if(Philostatus[philID].left==1 && Philostatus[philID].right==0){

if(philID==(n-1)){

if(ForkAvil[philID].taken==0){

ForkAvil[philID].taken = Philostatus[philID].right = 1;

cout<<"Fork "<<philID+1<<" taken by philosopher "<<philID+1<<"\n";

}else{

cout<<"Philosopher "<<philID+1<<" is waiting for fork "<<philID+1<<"\n";

}

}else{

int dupphilID = philID;

philID-=1;

if(philID== -1)

philID=(n-1);

if(ForkAvil[philID].taken == 0){

ForkAvil[philID].taken = Philostatus[dupphilID].right = 1;

cout<<"Fork "<<philID+1<<" taken by Philosopher "<<dupphilID+1<<"\n";

}else{

cout<<"Philosopher "<<dupphilID+1<<" is waiting for Fork "<<philID+1<<"\n";

}

}

}

else if(Philostatus[philID].left==0){

if(philID==(n-1)){

if(ForkAvil[philID-1].taken==0){

ForkAvil[philID-1].taken = Philostatus[philID].left = 1;

cout<<"Fork "<<philID<<" taken by philosopher "<<philID+1<<"\n";

}else{

cout<<"Philosopher "<<philID+1<<" is waiting for fork "<<philID<<"\n";

}

}else{

if(ForkAvil[philID].taken == 0){

ForkAvil[philID].taken = Philostatus[philID].left = 1;

cout<<"Fork "<<philID+1<<" taken by Philosopher "<<philID+1<<"\n";

}else{

cout<<"Philosopher "<<philID+1<<" is waiting for Fork "<<philID+1<<"\n";

}

}

}else{}

}

int main(){

for(i=0;i<n;i++)

ForkAvil[i].taken=Philostatus[i].left=Philostatus[i].right=0;

while(compltedPhilo<n){

for(i=0;i<n;i++)

goForDinner(i);

cout<<"\nTill now num of philosophers completed dinner are "<<compltedPhilo<<"\n\n";

}

return 0;

}