1) Implement the above code and paste the screen shot of the output.

CODE:

#include <stdio.h>

#define n 4

int completedPhilo = 0, i;

struct fork {

int taken;

} ForkAvil[n];

struct philosopher {

int left;

int right;

} Philostatus[n];

void goForDinner(int philID) {

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

printf("Philosopher %d already completed his dinner\n", philID + 1);

}

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

printf("Philosopher %d completed his dinner\n", philID + 1);

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

int otherFork = (philID - 1 + n) % n;

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

printf("Philosopher %d released fork %d and fork %d\n", philID + 1, philID + 1, otherFork + 1);

completedPhilo++;

}

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

int otherFork = philID == n - 1 ? philID : (philID - 1 + n) % n;

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

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

printf("Fork %d taken by philosopher %d\n", otherFork + 1, philID + 1);

} else {

printf("Philosopher %d is waiting for fork %d\n", philID + 1, otherFork + 1);

}

}

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

int otherFork = philID == n - 1 ? philID - 1 : philID;

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

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

printf("Fork %d taken by philosopher %d\n", otherFork + 1, philID + 1);

} else {

printf("Philosopher %d is waiting for fork %d\n", philID + 1, otherFork + 1);

}

}

}

int main() {

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

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

}

while (completedPhilo < n) {

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

goForDinner(i);

}

printf("\nTill now, number of philosophers who completed dinner: %d\n\n", completedPhilo);

}

return 0;

}

**OUTPUT**

