Shayan
DT-22037

OS LAB 06

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

Solution:

```
#include <stdio.h>

#define n 4 // Number of philosophers

int completedPhilo = 0, i;

struct fork
{
    int taken;
} ForkAvail[n];

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

void goForDinner(int philID)
{
    if (Philostatus[philID].left == 10 && Philostatus[philID].right == 10)
    {
        // Already completed
        printf("Philosopher %d already completed his dinner\n", philID + 1);
    }
    else if (Philostatus[philID].left == 1 && Philostatus[philID].right == 1)
    {
        // Completed dinner now
        printf("Philosopher %d completed his dinner\n", philID + 1);
        Philostatus[philID].left = Philostatus[philID].right = 10;

        int otherFork = philID - 1;
        if (otherFork == -1)
            otherFork = n - 1;

        ForkAvail[philID].taken = ForkAvail[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)
    {
        // Has left, trying right
        if (philID == n - 1)
        {
            if (ForkAvail[philID].taken == 0)
            {
                ForkAvail[philID].taken = 1;
                Philostatus[philID].right = 1;
                printf("Fork %d taken by philosopher %d\n", philID + 1, philID + 1);
            }
            else
            {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID +
1);
            }
        }
        else
        {
            int dupPhilID = philID;
            philID -= 1;
            if (philID == -1)
                philID = n - 1;

            if (ForkAvail[philID].taken == 0)
            {
                ForkAvail[philID].taken = 1;
                Philostatus[dupPhilID].right = 1;
                printf("Fork %d taken by philosopher %d\n", philID + 1, dupPhilID + 1);
            }
            else
            {
                printf("Philosopher %d is waiting for fork %d\n", dupPhilID + 1, philID
+ 1);
            }
        }
    }
    else if (Philostatus[philID].left == 0)
    {
        // Nothing taken yet
        if (philID == n - 1)
        {
            if (ForkAvail[philID - 1].taken == 0)
            {
                ForkAvail[philID - 1].taken = 1;
                Philostatus[philID].left = 1;
                printf("Fork %d taken by philosopher %d\n", philID, philID + 1);
```

```
            }
            else
            {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID);
            }
        }
        else
        {
            if (ForkAvail[philID].taken == 0)
            {
                ForkAvail[philID].taken = 1;
                Philostatus[philID].left = 1;
                printf("Fork %d taken by philosopher %d\n", philID + 1, philID + 1);
            }
            else
            {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID +
1);
            }
        }
    }
}

int main()
{
    // Initialize forks and philosophers
    for (i = 0; i < n; i++)
    {
        ForkAvail[i].taken = 0;
        Philostatus[i].left = 0;
        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;
}
```

Shayan
DT-22037

```
Fork 1 taken by philosopher 1
Fork 2 taken by philosopher 2
Fork 3 taken by philosopher 3
Philosopher 4 is waiting for fork 3

Till now number of philosophers who completed dinner: 0

Fork 4 taken by philosopher 1
Philosopher 2 is waiting for fork 1
Philosopher 3 is waiting for fork 2
Philosopher 4 is waiting for fork 3

Till now number of philosophers who completed dinner: 0

Philosopher 1 completed his dinner
Philosopher 1 released fork 1 and fork 4
Fork 1 taken by philosopher 2
Philosopher 3 is waiting for fork 2
Philosopher 4 is waiting for fork 3

Till now number of philosophers who completed dinner: 1

Philosopher 1 already completed his dinner
Philosopher 2 completed his dinner
Philosopher 2 released fork 2 and fork 1
Fork 2 taken by philosopher 3
Philosopher 4 is waiting for fork 3

Till now number of philosophers who completed dinner: 2
```

```
Philosopher 1 already completed his dinner
Philosopher 2 already completed his dinner
Philosopher 3 completed his dinner
Philosopher 3 released fork 3 and fork 2
Fork 3 taken by philosopher 4

Till now number of philosophers who completed dinner: 3

Philosopher 1 already completed his dinner
Philosopher 2 already completed his dinner
Philosopher 3 already completed his dinner
Fork 4 taken by philosopher 4
Fork 4 taken by philosopher 4

Till now number of philosophers who completed dinner: 3

Philosopher 1 already completed his dinner
Philosopher 2 already completed his dinner
Philosopher 3 already completed his dinner
Philosopher 4 completed his dinner
Philosopher 4 released fork 4 and fork 3

Till now number of philosophers who completed dinner: 4
```