OS LAB 06

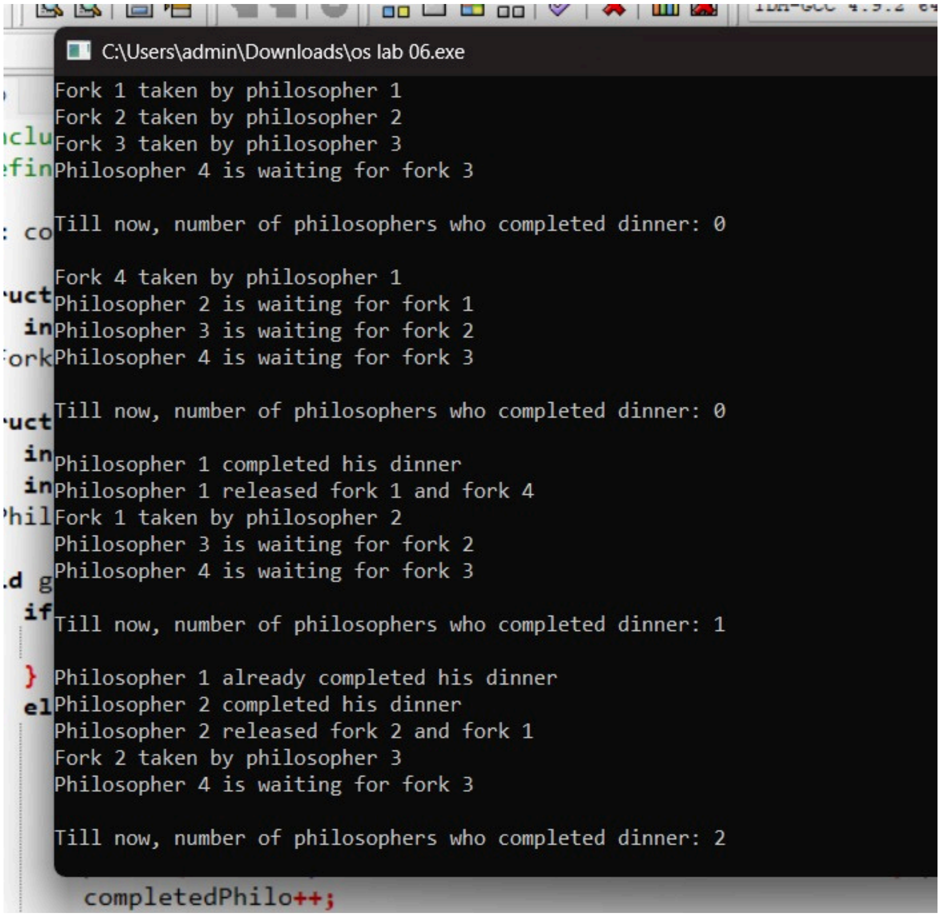
1) Implement the above code and paste the screen shot of the output.

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;
}
```

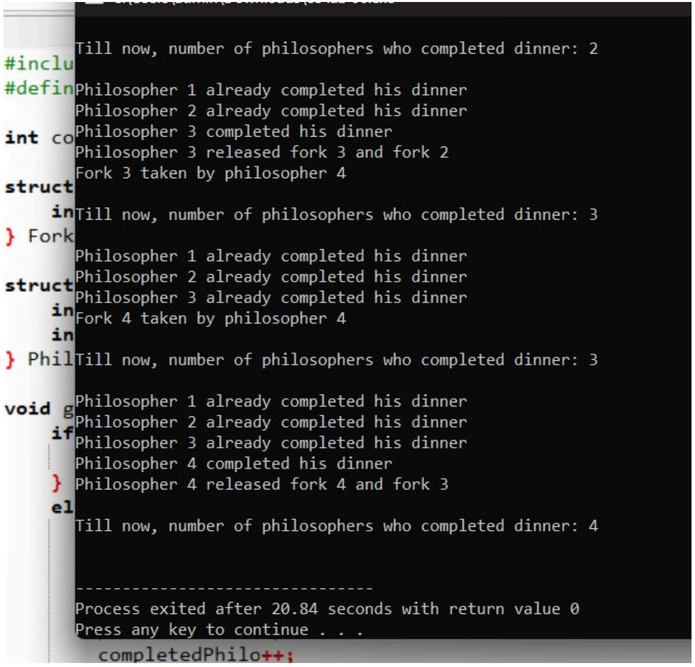
OS LAB 06

OUTPUT



```
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
```

```
completedPhilo++;
```



```
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

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


--------------------------------
Process exited after 20.84 seconds with return value 0
Press any key to continue . . .
```

```
completedPhilo++;
```