OS-Lab

Lab#10

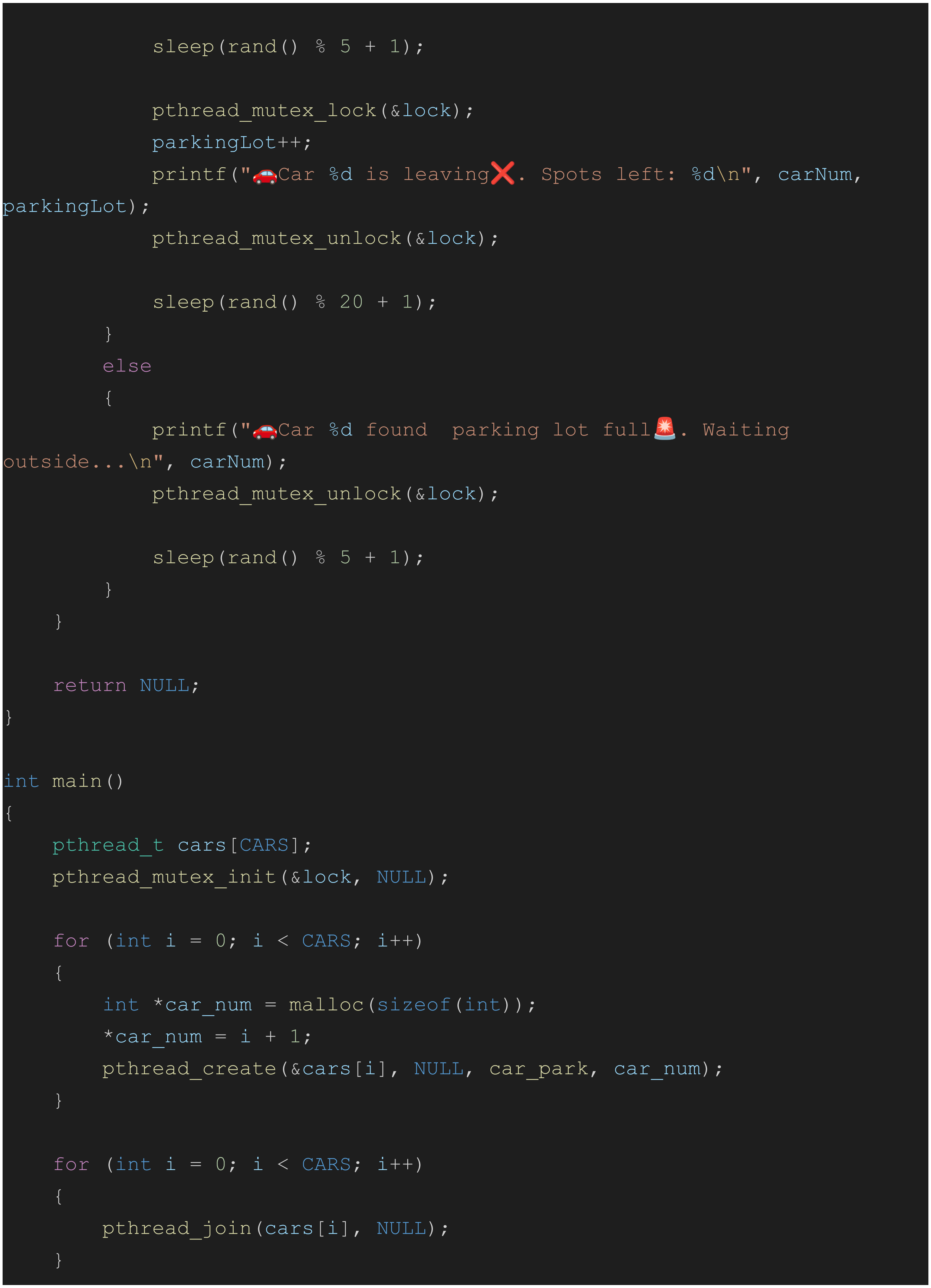
Name: Anas Altaf

Roll.no: 22F-3639

C Codes:

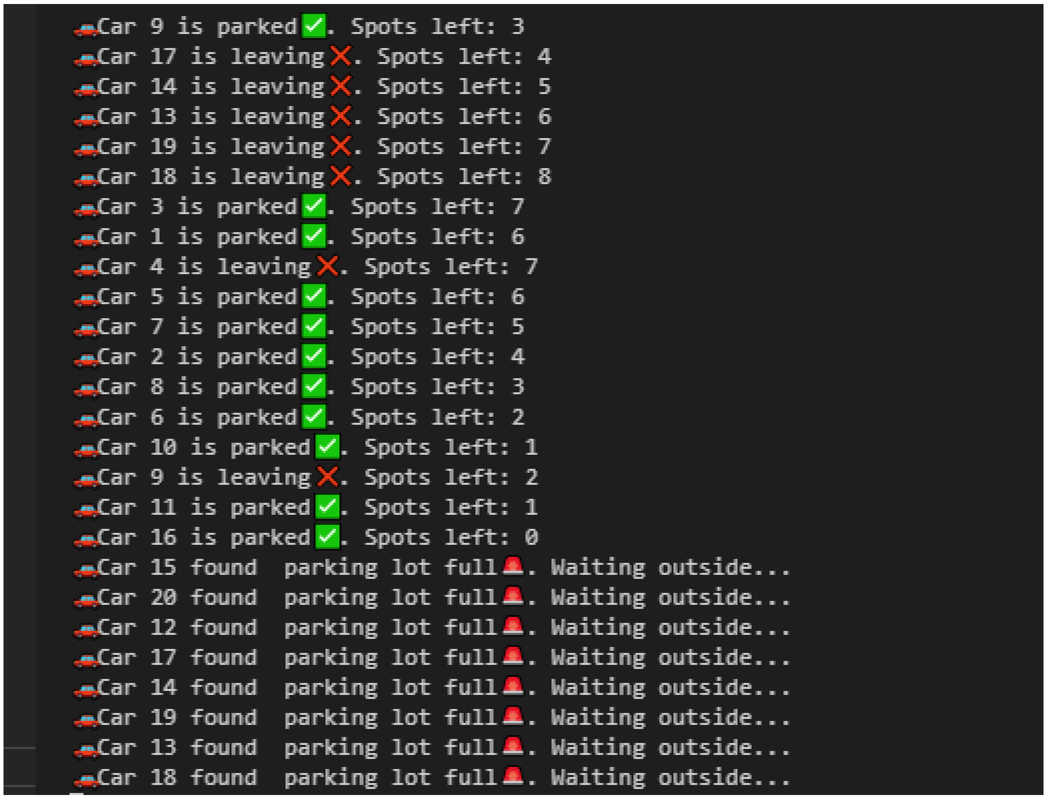
T1:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

#define MAX_SPOTS 10
#define CARS 20

pthread_mutex_t lock;
int parkingLot = MAX_SPOTS;

void *car_park(void *carNumber)
{
    int carNum = *(int *)carNumber;

    while (1)
    {
        pthread_mutex_lock(&lock);
        if (parkingLot > 0)
        {
            parkingLot--;
            printf("🚗Car %d is parked✅. Spots left: %d\n", carNum,
parkingLot);
            pthread_mutex_unlock(&lock);
```



```c
            sleep(rand() % 5 + 1);

            pthread_mutex_lock(&lock);
            parkingLot++;
            printf("🚗Car %d is leaving❌. Spots left: %d\n", carNum, parkingLot);
            pthread_mutex_unlock(&lock);

            sleep(rand() % 20 + 1);
        }
        else
        {
            printf("🚗Car %d found  parking lot full🚨. Waiting outside...\n", carNum);
            pthread_mutex_unlock(&lock);

            sleep(rand() % 5 + 1);
        }
    }

    return NULL;
}

int main()
{
    pthread_t cars[CARS];
    pthread_mutex_init(&lock, NULL);

    for (int i = 0; i < CARS; i++)
    {
        int *car_num = malloc(sizeof(int));
        *car_num = i + 1;
        pthread_create(&cars[i], NULL, car_park, car_num);
    }

    for (int i = 0; i < CARS; i++)
    {
        pthread_join(cars[i], NULL);
    }
    pthread_mutex_destroy(&lock);
    return 0;
}
```

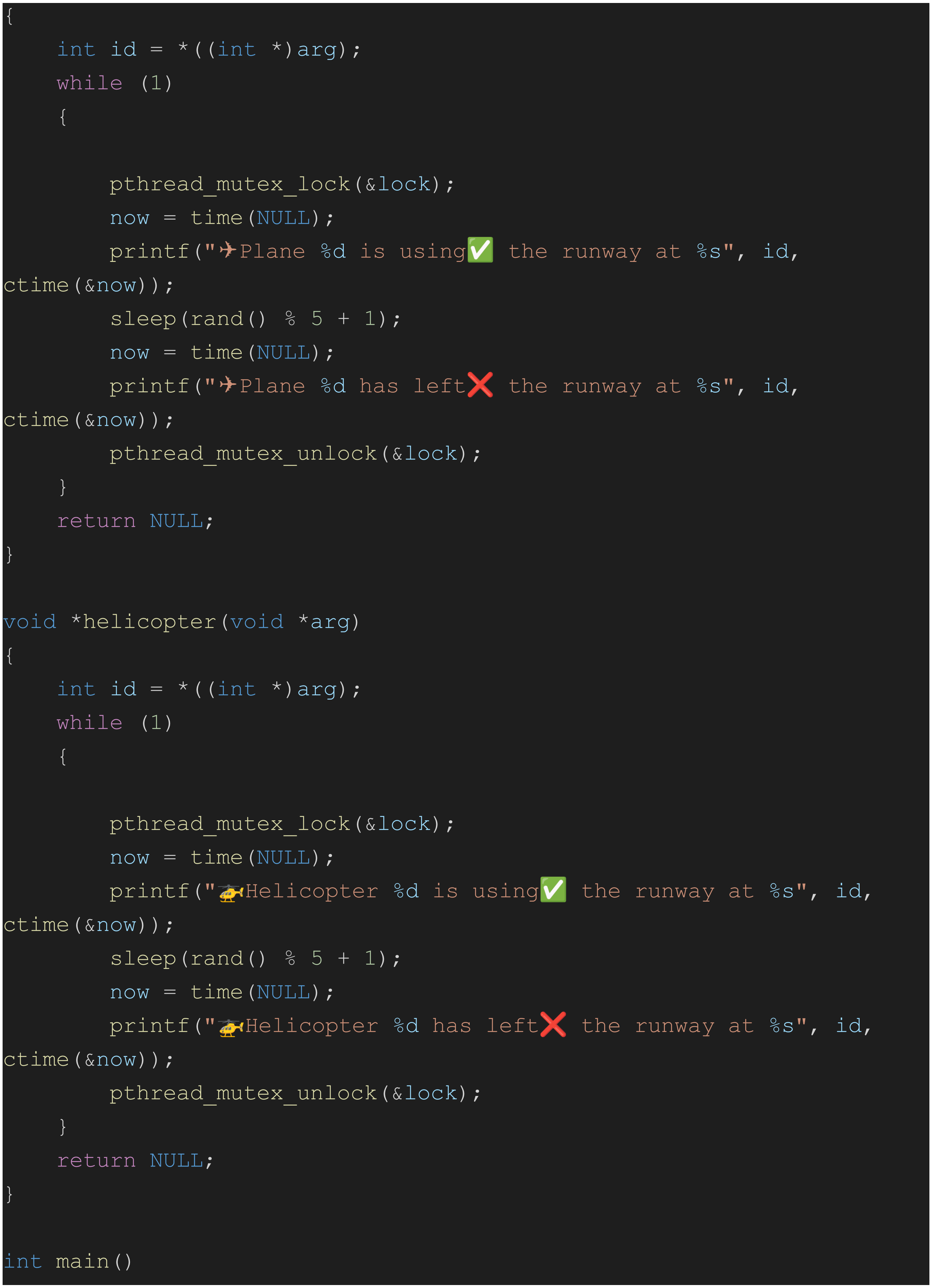
Output:



```
🚗Car 9 is parked✅. Spots left: 3
🚗Car 17 is leaving❌. Spots left: 4
🚗Car 14 is leaving❌. Spots left: 5
🚗Car 13 is leaving❌. Spots left: 6
🚗Car 19 is leaving❌. Spots left: 7
🚗Car 18 is leaving❌. Spots left: 8
🚗Car 3 is parked✅. Spots left: 7
🚗Car 1 is parked✅. Spots left: 6
🚗Car 4 is leaving❌. Spots left: 7
🚗Car 5 is parked✅. Spots left: 6
🚗Car 7 is parked✅. Spots left: 5
🚗Car 2 is parked✅. Spots left: 4
🚗Car 8 is parked✅. Spots left: 3
🚗Car 6 is parked✅. Spots left: 2
🚗Car 10 is parked✅. Spots left: 1
🚗Car 9 is leaving❌. Spots left: 2
🚗Car 11 is parked✅. Spots left: 1
🚗Car 16 is parked✅. Spots left: 0
🚗Car 15 found  parking lot full🚨. Waiting outside...
🚗Car 20 found  parking lot full🚨. Waiting outside...
🚗Car 12 found  parking lot full🚨. Waiting outside...
🚗Car 17 found  parking lot full🚨. Waiting outside...
🚗Car 14 found  parking lot full🚨. Waiting outside...
🚗Car 19 found  parking lot full🚨. Waiting outside...
🚗Car 13 found  parking lot full🚨. Waiting outside...
🚗Car 18 found  parking lot full🚨. Waiting outside...
```

T2:

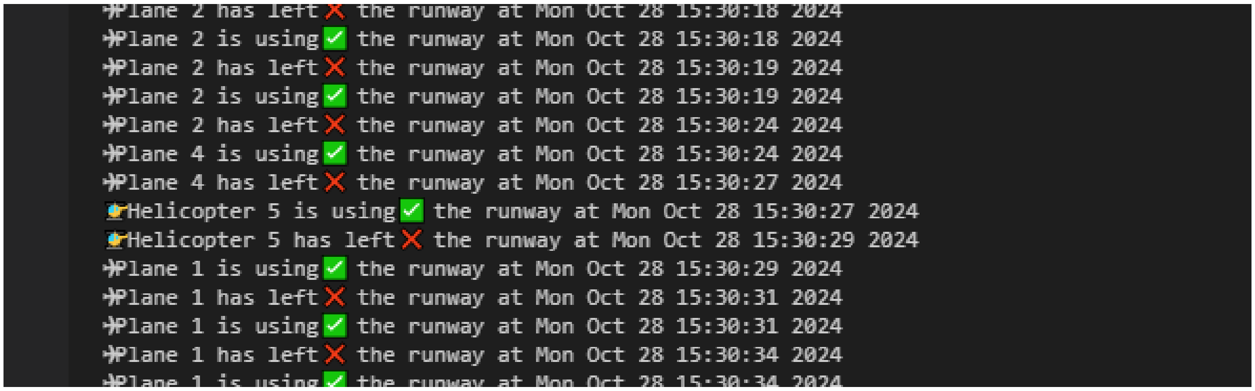
```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <time.h>

#define FLYABLES 5

pthread_mutex_t lock;
time_t now;
void *plane(void *arg)
```



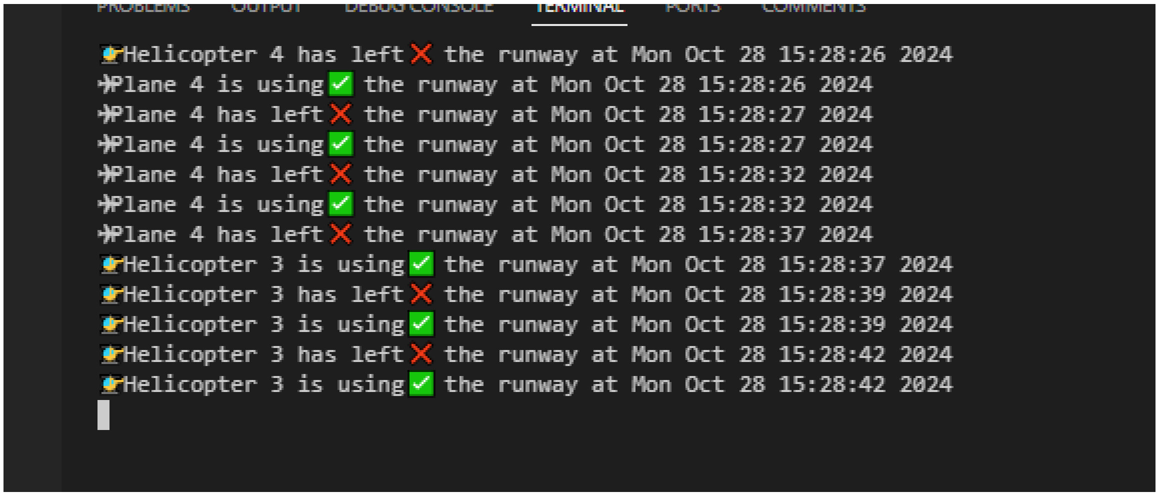
```
{
    int id = *((int *)arg);
    while (1)
    {

        pthread_mutex_lock(&lock);
        now = time(NULL);
        printf("✈Plane %d is using✅ the runway at %s", id,
ctime(&now));
        sleep(rand() % 5 + 1);
        now = time(NULL);
        printf("✈Plane %d has left❌ the runway at %s", id,
ctime(&now));
        pthread_mutex_unlock(&lock);
    }
    return NULL;
}

void *helicopter(void *arg)
{
    int id = *((int *)arg);
    while (1)
    {

        pthread_mutex_lock(&lock);
        now = time(NULL);
        printf("🚁Helicopter %d is using✅ the runway at %s", id,
ctime(&now));
        sleep(rand() % 5 + 1);
        now = time(NULL);
        printf("🚁Helicopter %d has left❌ the runway at %s", id,
ctime(&now));
        pthread_mutex_unlock(&lock);
    }
    return NULL;
}

int main()
```

```
{
    pthread_t planes[FLYABLES];
    pthread_t helicopters[FLYABLES];

    pthread_mutex_init(&lock, NULL);

    for (int i = 0; i < FLYABLES; i++)
    {
        int *id = malloc(sizeof(int));
        *id = i + 1;
        pthread_create(&planes[i], NULL, plane, id);
        pthread_create(&helicopters[i], NULL, helicopter, id);
    }

    for (int i = 0; i < FLYABLES; i++)
    {
        pthread_join(planes[i], NULL);
        pthread_join(helicopters[i], NULL);
    }

    pthread_mutex_destroy(&lock);

    return 0;
}
```

Output:



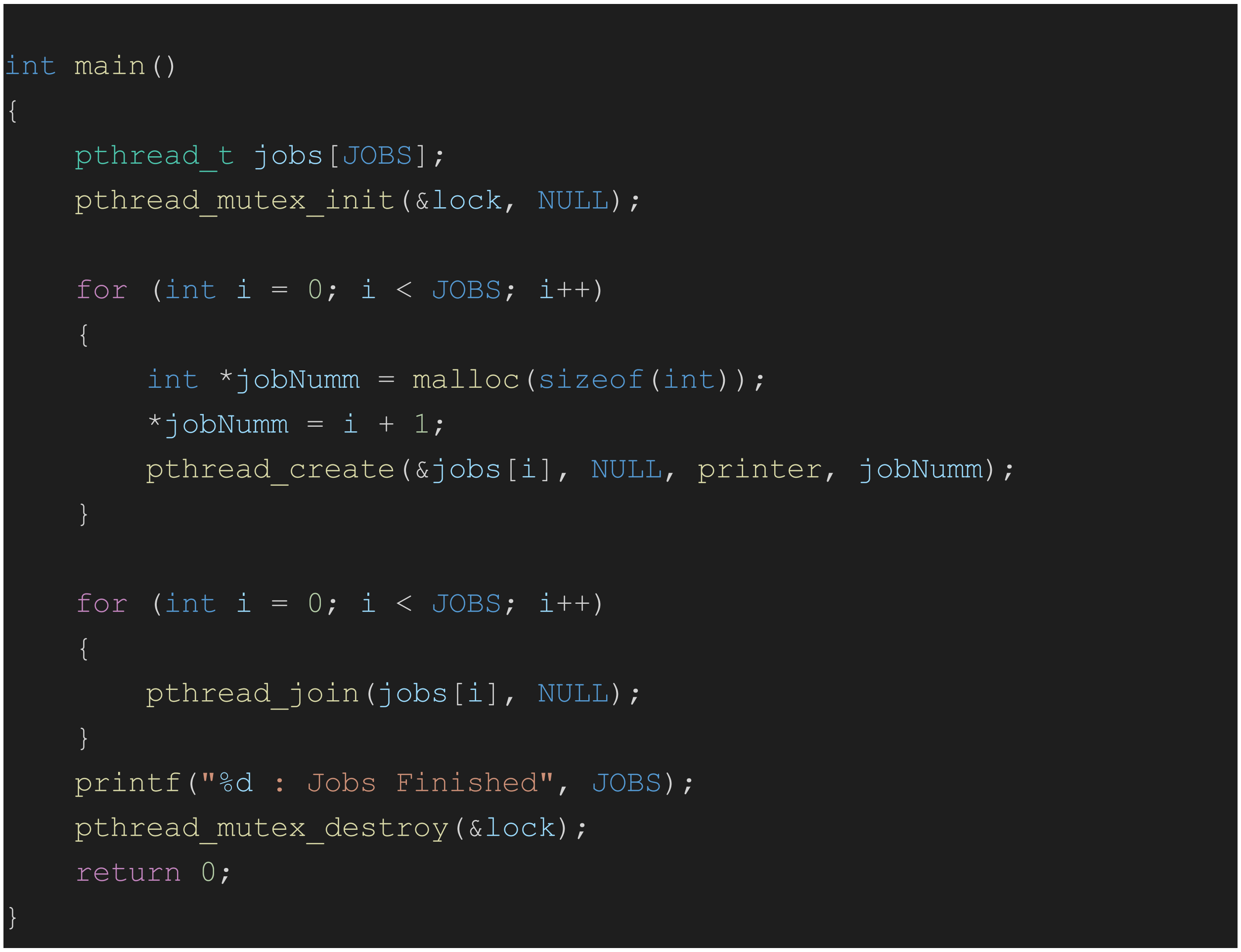
```
✈Plane 2 has left❌ the runway at Mon Oct 28 15:30:18 2024
✈Plane 2 is using✅ the runway at Mon Oct 28 15:30:18 2024
✈Plane 2 has left❌ the runway at Mon Oct 28 15:30:19 2024
✈Plane 2 is using✅ the runway at Mon Oct 28 15:30:19 2024
✈Plane 2 has left❌ the runway at Mon Oct 28 15:30:24 2024
✈Plane 4 is using✅ the runway at Mon Oct 28 15:30:24 2024
✈Plane 4 has left❌ the runway at Mon Oct 28 15:30:27 2024
🚁Helicopter 5 is using✅ the runway at Mon Oct 28 15:30:27 2024
🚁Helicopter 5 has left❌ the runway at Mon Oct 28 15:30:29 2024
✈Plane 1 is using✅ the runway at Mon Oct 28 15:30:29 2024
✈Plane 1 has left❌ the runway at Mon Oct 28 15:30:31 2024
✈Plane 1 is using✅ the runway at Mon Oct 28 15:30:31 2024
✈Plane 1 has left❌ the runway at Mon Oct 28 15:30:34 2024
✈Plane 1 is using✅ the runway at Mon Oct 28 15:30:34 2024
```



```
🚁Helicopter 4 has left❌ the runway at Mon Oct 28 15:28:26 2024
✈Plane 4 is using✅ the runway at Mon Oct 28 15:28:26 2024
✈Plane 4 has left❌ the runway at Mon Oct 28 15:28:27 2024
✈Plane 4 is using✅ the runway at Mon Oct 28 15:28:27 2024
✈Plane 4 has left❌ the runway at Mon Oct 28 15:28:32 2024
✈Plane 4 is using✅ the runway at Mon Oct 28 15:28:32 2024
✈Plane 4 has left❌ the runway at Mon Oct 28 15:28:37 2024
🚁Helicopter 3 is using✅ the runway at Mon Oct 28 15:28:37 2024
🚁Helicopter 3 has left❌ the runway at Mon Oct 28 15:28:39 2024
🚁Helicopter 3 is using✅ the runway at Mon Oct 28 15:28:39 2024
🚁Helicopter 3 has left❌ the runway at Mon Oct 28 15:28:42 2024
🚁Helicopter 3 is using✅ the runway at Mon Oct 28 15:28:42 2024
```

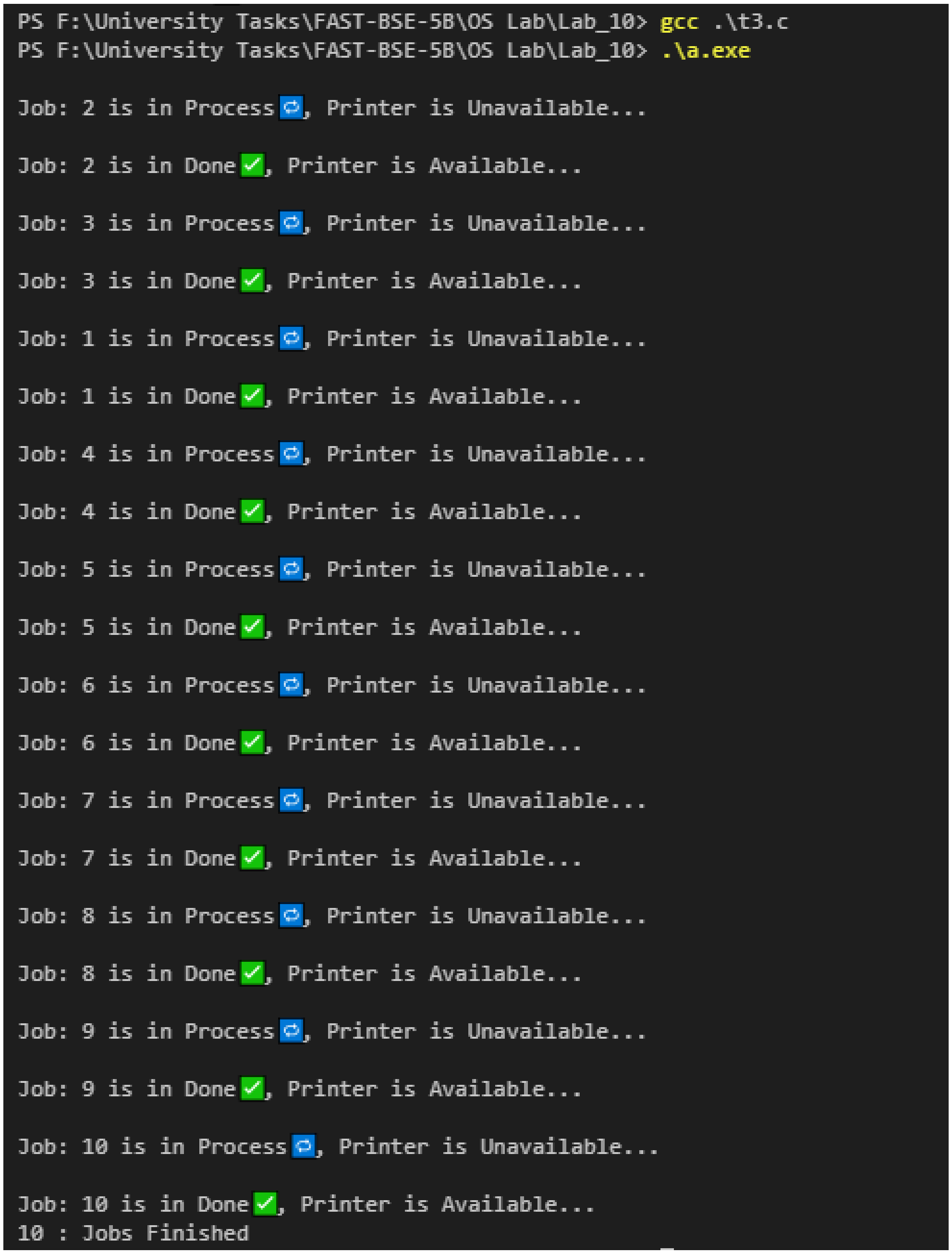
T3:

```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

#define JOBS 10

pthread_mutex_t lock;

void *printer(void *jobNum)
{
    int job = *(int *)jobNum;

    pthread_mutex_lock(&lock);

    printf("\nJob: %d is in Process🔁, Printer is Unavailable...\n",
job);

    sleep(rand() % 5 + 1);
    printf("\nJob: %d is in Done✅, Printer is Available...\n", job);
    pthread_mutex_unlock(&lock);

    return NULL;
}
```



```
int main()
{
    pthread_t jobs[JOBS];
    pthread_mutex_init(&lock, NULL);

    for (int i = 0; i < JOBS; i++)
    {
        int *jobNumm = malloc(sizeof(int));
        *jobNumm = i + 1;
        pthread_create(&jobs[i], NULL, printer, jobNumm);
    }

    for (int i = 0; i < JOBS; i++)
    {
        pthread_join(jobs[i], NULL);
    }
    printf("%d : Jobs Finished", JOBS);
    pthread_mutex_destroy(&lock);
    return 0;
}
```

Output:



```
PS F:\University Tasks\FAST-BSE-5B\OS Lab\Lab_10> gcc .\t3.c
PS F:\University Tasks\FAST-BSE-5B\OS Lab\Lab_10> .\a.exe

Job: 2 is in Process🔁, Printer is Unavailable...

Job: 2 is in Done✅, Printer is Available...

Job: 3 is in Process🔁, Printer is Unavailable...

Job: 3 is in Done✅, Printer is Available...

Job: 1 is in Process🔁, Printer is Unavailable...

Job: 1 is in Done✅, Printer is Available...

Job: 4 is in Process🔁, Printer is Unavailable...

Job: 4 is in Done✅, Printer is Available...

Job: 5 is in Process🔁, Printer is Unavailable...

Job: 5 is in Done✅, Printer is Available...

Job: 6 is in Process🔁, Printer is Unavailable...

Job: 6 is in Done✅, Printer is Available...

Job: 7 is in Process🔁, Printer is Unavailable...

Job: 7 is in Done✅, Printer is Available...

Job: 8 is in Process🔁, Printer is Unavailable...

Job: 8 is in Done✅, Printer is Available...

Job: 9 is in Process🔁, Printer is Unavailable...

Job: 9 is in Done✅, Printer is Available...

Job: 10 is in Process🔁, Printer is Unavailable...

Job: 10 is in Done✅, Printer is Available...
10 : Jobs Finished
```