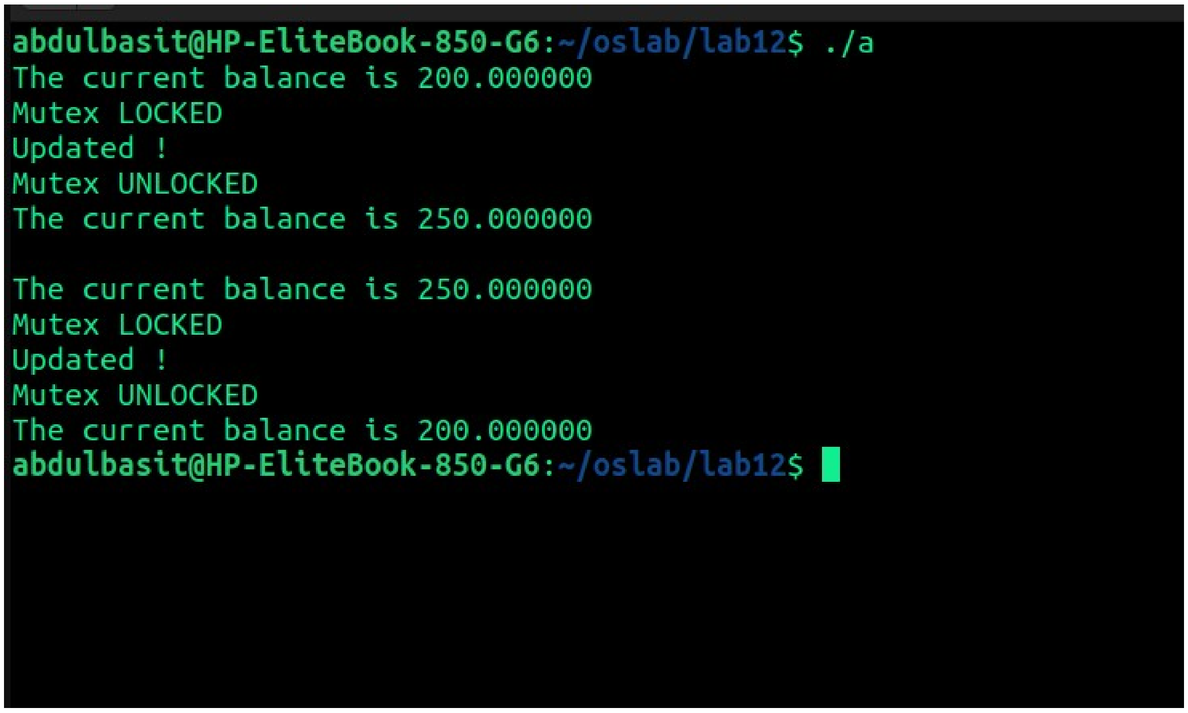
I233018
Abdul Basit
LAB12

Q1:
CODE:
```
#include<unistd.h>
#include <pthread.h>
#include <stdio.h>

float BANK_BALANCE=200.00;
pthread_mutex_t lock;

void* increaseBalance(void * args)
{
	printf("The current balance is %f",BANK_BALANCE);
	pthread_mutex_lock(&lock);
	printf("\nMutex LOCKED");
	BANK_BALANCE+=50.0;
	printf("\nUpdated !");
	pthread_mutex_unlock(&lock);
	printf("\nMutex UNLOCKED");
	printf("\nThe current balance is %f\n",BANK_BALANCE);

}

void* decreaseBalance(void * args)
{
	sleep(1);
	pthread_mutex_lock(&lock);
	printf("\nThe current balance is %f",BANK_BALANCE);
	printf("\nMutex LOCKED");
	BANK_BALANCE-=50.0;
	printf("\nUpdated !");
	pthread_mutex_unlock(&lock);
	printf("\nMutex UNLOCKED");
	printf("\nThe current balance is %f\n",BANK_BALANCE);
}

int main ()
{
	pthread_t t1,t2;
	pthread_mutex_init(&lock,NULL);

	pthread_create(&t1,NULL,increaseBalance,NULL);
	pthread_create(&t2,NULL,decreaseBalance,NULL);

	pthread_join(t1,NULL);
	pthread_join(t2,NULL);
	return 0;
}
```



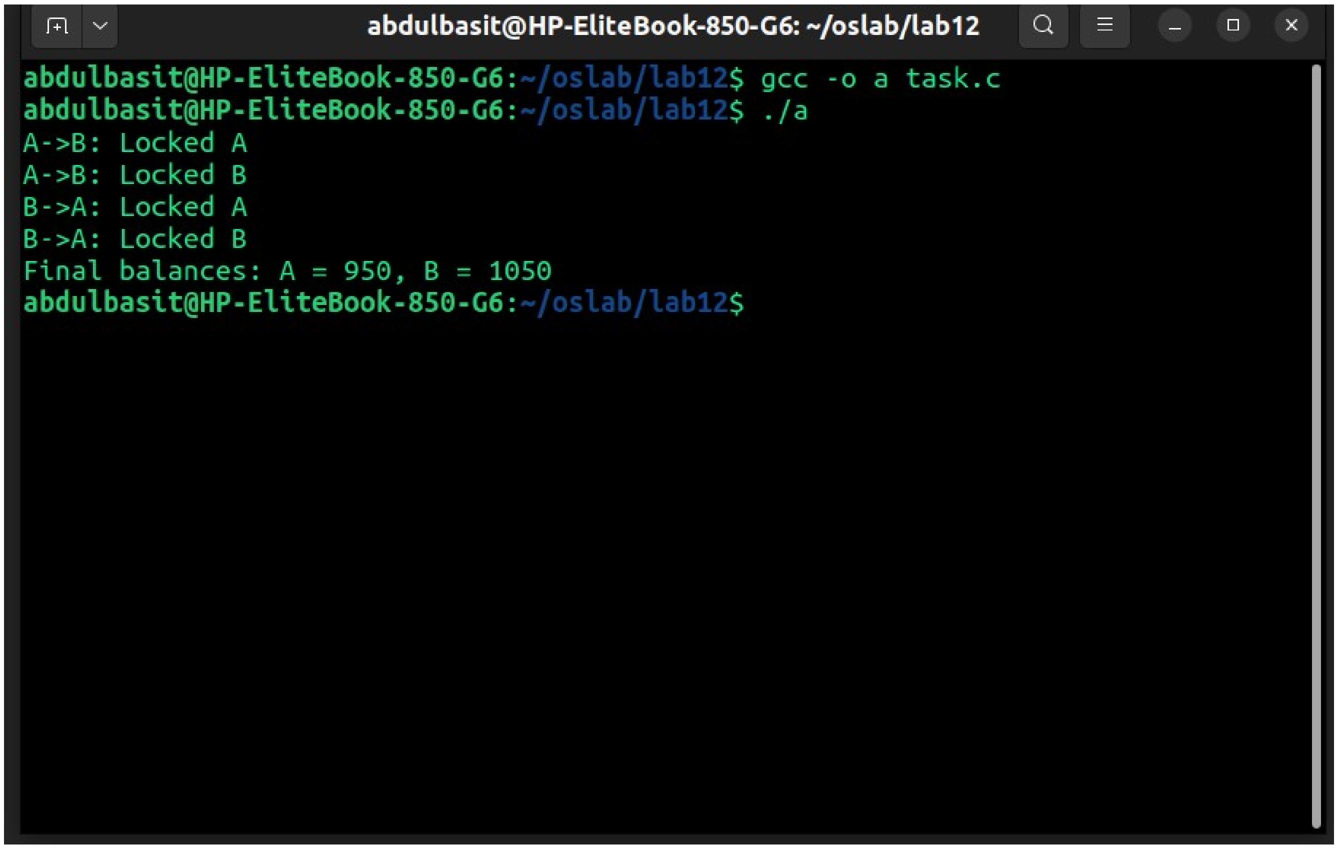
```
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$ ./a
The current balance is 200.000000
Mutex LOCKED
Updated !
Mutex UNLOCKED
The current balance is 250.000000

The current balance is 250.000000
Mutex LOCKED
Updated !
Mutex UNLOCKED
The current balance is 200.000000
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$ 
```

Q2;
CODE:

```
//Simulating deadlock

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

int balanceA = 1000;
int balanceB = 1000;

pthread_mutex_t lockA = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t lockB = PTHREAD_MUTEX_INITIALIZER;

void* transferAtoB(void* arg)
{
	pthread_mutex_lock(&lockA);
	printf("A->B: Locked A\n");
	sleep(1);
	pthread_mutex_lock(&lockB);
	printf("A->B: Locked B\n");
	balanceA -= 100;
	balanceB += 100;
	pthread_mutex_unlock(&lockB);
```

```
        pthread_mutex_unlock(&lockA);
        return NULL;

}


void* transferBtoA(void* arg)
{
        pthread_mutex_lock(&lockB);
        printf("B->A: Locked B\n");
        sleep(1);
        pthread_mutex_lock(&lockA);
        printf("B->A: Locked A\n");
        balanceB -= 50;
        balanceA += 50;
        pthread_mutex_unlock(&lockA);
        pthread_mutex_unlock(&lockB);
        return NULL;

}


//without deadlock

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


int balanceA = 1000;
int balanceB = 1000;


pthread_mutex_t lockA = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t lockB = PTHREAD_MUTEX_INITIALIZER;


void* transferAtoB(void* arg)
{
        pthread_mutex_lock(&lockA);
        printf("A->B: Locked A\n");
        sleep(1);
        pthread_mutex_lock(&lockB);
        printf("A->B: Locked B\n");
        balanceA -= 100;
        balanceB += 100;
        pthread_mutex_unlock(&lockB);
        pthread_mutex_unlock(&lockA);
        return NULL;
```

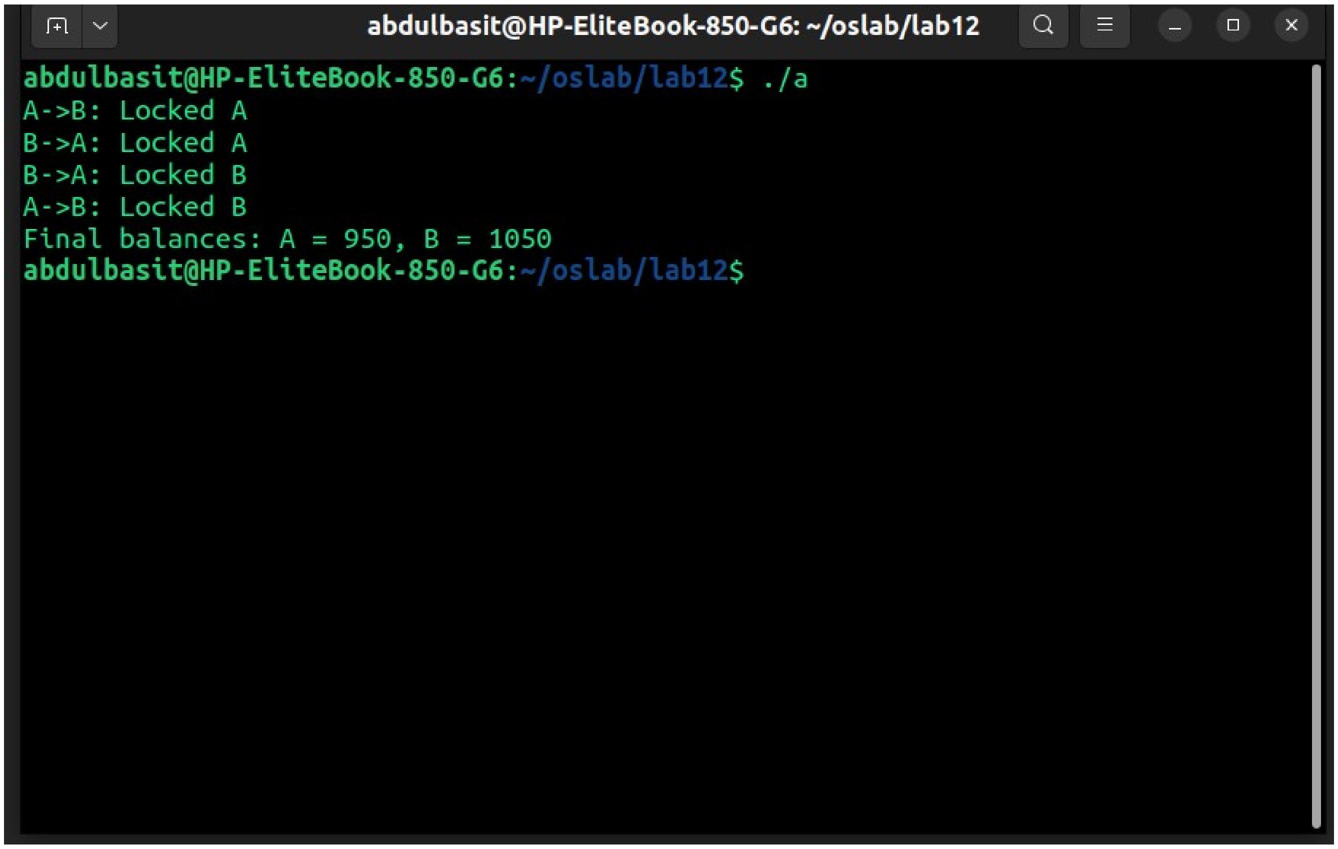
```
}


void* transferBtoA(void* arg)
{
        pthread_mutex_lock(&lockA);
        printf("B->A: Locked A\n");
        sleep(1);
        pthread_mutex_lock(&lockB);
        printf("B->A: Locked B\n");
        balanceB -= 50;
        balanceA += 50;
        pthread_mutex_unlock(&lockA);
        pthread_mutex_unlock(&lockB);
        return NULL;

}


int main()
{
        pthread_t t1, t2;
        pthread_create(&t1, NULL, transferAtoB, NULL);
        pthread_create(&t2, NULL, transferBtoA, NULL);
        pthread_join(t1, NULL);
        pthread_join(t2, NULL);
        printf("Final balances: A = %d, B = %d\n", balanceA, balanceB);

return 0;

}
```



```
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$ gcc -o a task.c
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$ ./a
A->B: Locked A
A->B: Locked B
B->A: Locked A
B->A: Locked B
Final balances: A = 950, B = 1050
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$
```

Q3:
CODE:

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

int balanceA = 1000;
int balanceB = 1000;

pthread_mutex_t lockA = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t lockB = PTHREAD_MUTEX_INITIALIZER;

void* transferAtoB(void* arg) {
    while (pthread_mutex_trylock(&lockA)) {
        sleep(1);
    }

    printf("A->B: Locked A\n");
    sleep(1);

    while (pthread_mutex_trylock(&lockB)) {
        sleep(1);
    }

    printf("A->B: Locked B\n");
    balanceA -= 100;
    balanceB += 100;

    pthread_mutex_unlock(&lockB);
    pthread_mutex_unlock(&lockA);

    return NULL;
}

void* transferBtoA(void* arg) {
    pthread_mutex_trylock(&lockA);

    printf("B->A: Locked A\n");
    sleep(1);

    pthread_mutex_trylock(&lockB);

    printf("B->A: Locked B\n");

    balanceB -= 50;
    balanceA += 50;

    pthread_mutex_unlock(&lockA);
    pthread_mutex_unlock(&lockB);

    return NULL;
}

int main() {
    pthread_t t1, t2;

    pthread_create(&t1, NULL, transferAtoB, NULL);
    pthread_create(&t2, NULL, transferBtoA, NULL);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    printf("Final balances: A = %d, B = %d\n", balanceA, balanceB);

    return 0;
}
```



```
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$ ./a
A->B: Locked A
B->A: Locked A
B->A: Locked B
A->B: Locked B
Final balances: A = 950, B = 1050
abdulbasit@HP-EliteBook-850-G6:~/oslab/lab12$
```