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

// thread deadlock example

int n=1000000;

pthread_mutex_t mutex0;
pthread_mutex_t mutex1;

int num0=0;
int num1=0;

void* run0(void * arg){
  for(int i=0;i<n;i++){
    pthread_mutex_lock(&mutex1);
    sleep(1);
    pthread_mutex_lock(&mutex0);

    num0++;
    num1--;
    printf("num1 -->%d\n",num1);
    pthread_mutex_unlock(&mutex0);
    pthread_mutex_unlock(&mutex1);
  }
  return NULL;
}

void* run1(void * arg){
  for(int i=0;i<n;i++){
    pthread_mutex_lock(&mutex1); // change mutex for deadlock yes/no
    sleep(1);
    pthread_mutex_lock(&mutex0);
    num0--;
    num1++;
    printf("num0 -->%d\n",num0);
    pthread_mutex_unlock(&mutex0);
    pthread_mutex_unlock(&mutex1);
  }
  return NULL;
}

int main(int arg, char* argv[])
{
  pthread_t thid0,thid1;
  pthread_mutex_init(&mutex0,NULL);
  pthread_mutex_init(&mutex1,NULL);

  int ris = pthread_create(&thid0, NULL, &run0, NULL);
  if (ris){
    printf("errore creazione thread\n");
    exit(-1);
  }
  ris = pthread_create(&thid1, NULL, &run1, NULL);
  if (ris){
    printf("errore creazione thread\n");
    exit(-1);
```

```
60    }
61    pthread_join(thid0, NULL);
62    pthread_join(thid1, NULL);
63    pthread_mutex_destroy(&mutex0);
64    pthread_mutex_destroy(&mutex1);
65  }
66
```