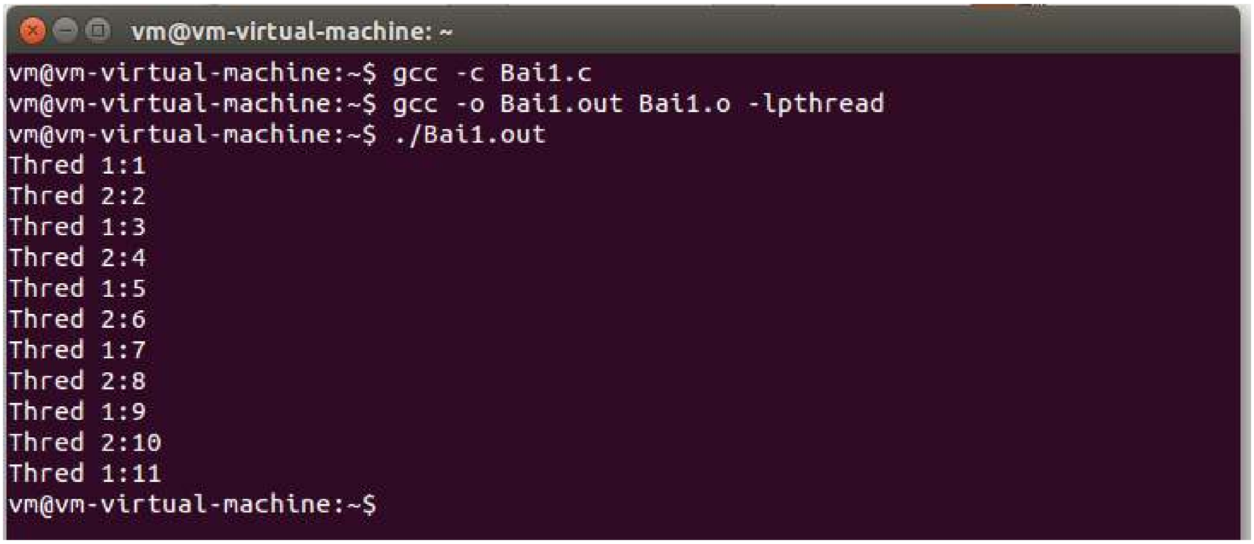
Câu 1:

```
//bai1
// C program to demonstrate working of Semaphores
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
sem_t mutex1,mutex2;
void *inle(void *arg)
{
    // wait
    int i;
    for(i=1;i<12;i+=2){
        sem_wait(&mutex2);
        printf("Thred 1:%d\n",i);
        sem_post(&mutex1);
    }

}
void *inchan(void *arg)
{
    // wait
    int i;
    for(i=2;i<11;i+=2){
        sem_wait(&mutex1);
        printf("Thred 2:%d\n",i);
        sem_post(&mutex2);
    }
}
int main()
{
    sem_init(&mutex1, 0, 0);
    sem_init(&mutex2, 0, 1);

    pthread_t t1, t2;
    pthread_create(&t1, NULL, inle, NULL);
    // sleep(2);
    pthread_create(&t2, NULL, inchan, NULL);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    sem_destroy(&mutex1);
sem_destroy(&mutex2);
    return 0;
}
```

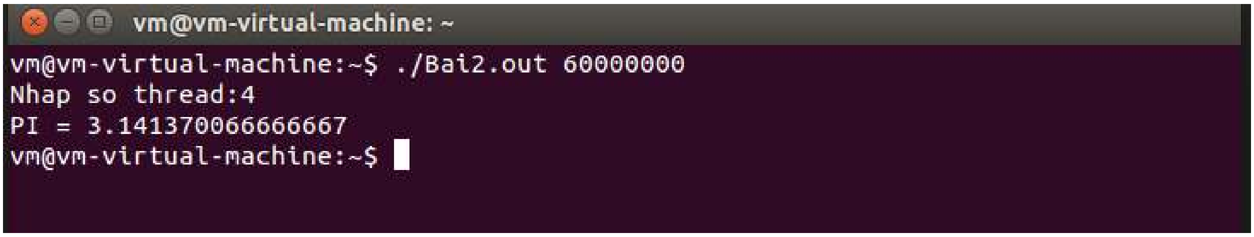


```
vm@vm-virtual-machine:~$ gcc -c Bai1.c
vm@vm-virtual-machine:~$ gcc -o Bai1.out Bai1.o -lpthread
vm@vm-virtual-machine:~$ ./Bai1.out
Thred 1:1
Thred 2:2
Thred 1:3
Thred 2:4
Thred 1:5
Thred 2:6
Thred 1:7
Thred 2:8
Thred 1:9
Thred 2:10
Thred 1:11
vm@vm-virtual-machine:~$
```

Câu 2:

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

```
#include <time.h>
#include <math.h>
#include <pthread.h>

long int total_point;
void *circle_point(void *param)
{
    int *pcount = (int *)param;
    int i;
    for (i = 0; i < total_point; i++)
    {
        double x = (double)rand() / (double)RAND_MAX;
        double y = (double)rand() / (double)RAND_MAX;
        double r = x * x + y * y;
        if (r <= 1)
            *pcount = *pcount + 1;
    }
    pthread_exit(0);
}

int main(int argc, char const *argv[])
{
    if (argc != 2)
    {
        printf("Error\n");
        return -1;
    }
    int NUM_THREAD;
    long int count_circle = 0;
    printf("Nhap so thread:");
    scanf("%d", &NUM_THREAD);
    sleep(1);
    pthread_t tid[4] = {0};
    int count[4] = {0};
    total_point = atoll(argv[1]) / NUM_THREAD;
    srand(time(NULL));
    int i;
    for (i = 0; i < NUM_THREAD; i++)
        pthread_create(&tid[i], NULL, circle_point, &count[i]);
    for (i = 0; i < NUM_THREAD; i++)
    {
        pthread_join(tid[i], NULL);
        count_circle += count[i];
    }
    double pi = 4.0 * (double)count_circle / (double)total_point / (double)NUM_THREAD;

    printf("PI = %17.15f\n", pi);
    return 0;
}
```

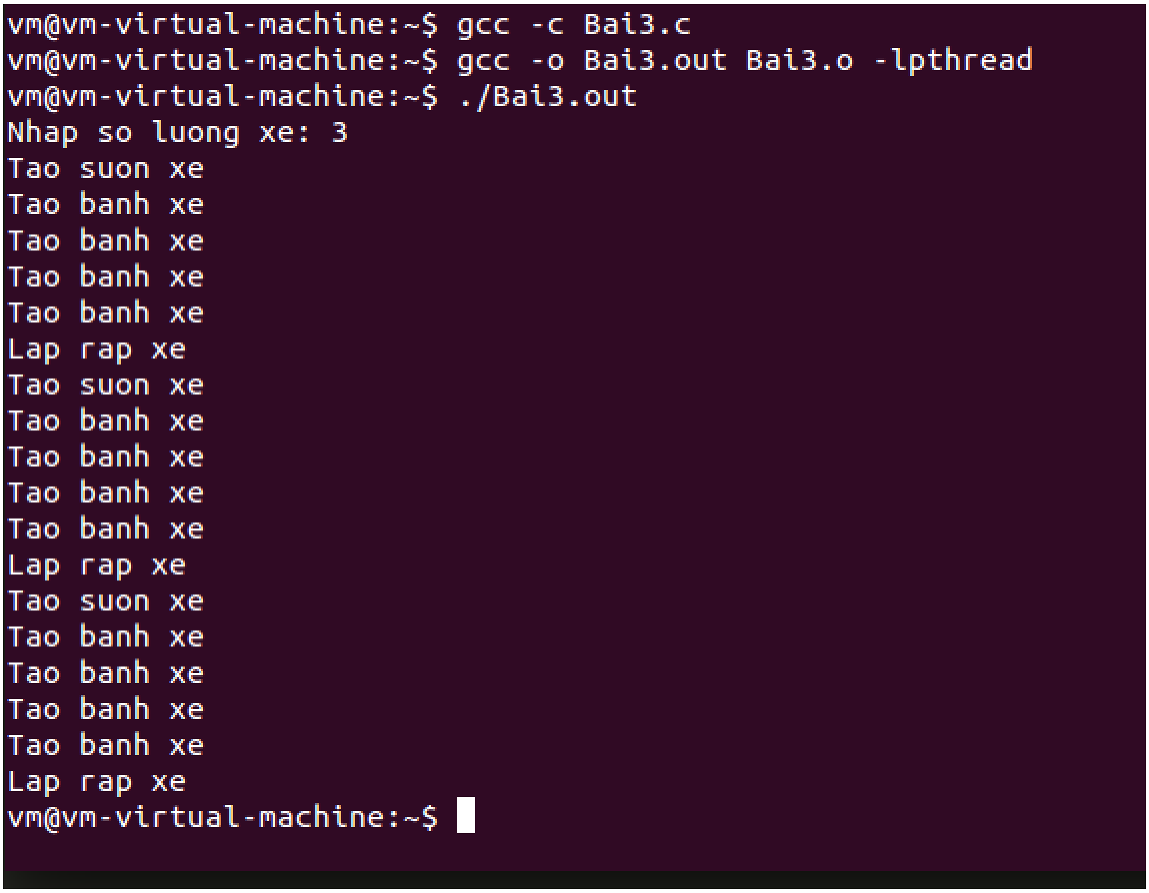


```
vm@vm-virtual-machine:~$ ./Bai2.out 60000000
Nhap so thread:4
PI = 3.141370066666667
vm@vm-virtual-machine:~$ 
```

Câu 3:

```
#include<stdio.h>
#include<semaphore.h>
#include<stdlib.h>
```

```
#include<pthread.h>

sem_t m1, m2, m3;
void *taosuon(void *argv){
        sem_wait(&m1);
        printf("Tao suon xe\n");
        sem_post(&m2);
}
void *taobanh(void *argv){
        int i;
        sem_wait(&m2);
        for(i=0;i<4;i++){
                printf("Tao banh xe\n");
        }
        sem_post(&m3);
}
void *lapxe(void *argv){
        sem_wait(&m3);
        printf("Lap rap xe\n");
        sem_post(&m1);
}

void main()
{
   int n,i;
   printf("Nhap so luong xe: ");
   scanf("%d", &n);

   sleep(2);
   for(i=0;i<n;i++){
      sem_init(&m1,0,1);
      sem_init(&m2,0,0);
      sem_init(&m3,0,0);

      pthread_t t1;
      pthread_t t2;
      pthread_t t3;
      pthread_create(&t1, NULL, taosuon, NULL);
      pthread_create(&t2, NULL, taobanh, NULL);
      pthread_create(&t3, NULL, lapxe, NULL);
      pthread_join(t1, NULL);
      pthread_join(t2, NULL);
      pthread_join(t3, NULL);
   }

           sem_destroy(&m1);
           sem_destroy(&m2);
           sem_destroy(&m3);
}
```



```
vm@vm-virtual-machine:~$ gcc -c Bai3.c
vm@vm-virtual-machine:~$ gcc -o Bai3.out Bai3.o -lpthread
vm@vm-virtual-machine:~$ ./Bai3.out
Nhap so luong xe: 3
Tao suon xe
Tao banh xe
Tao banh xe
Tao banh xe
Tao banh xe
Lap rap xe
Tao suon xe
Tao banh xe
Tao banh xe
Tao banh xe
Tao banh xe
Lap rap xe
Tao suon xe
Tao banh xe
Tao banh xe
Tao banh xe
Tao banh xe
Lap rap xe
vm@vm-virtual-machine:~$ 
```

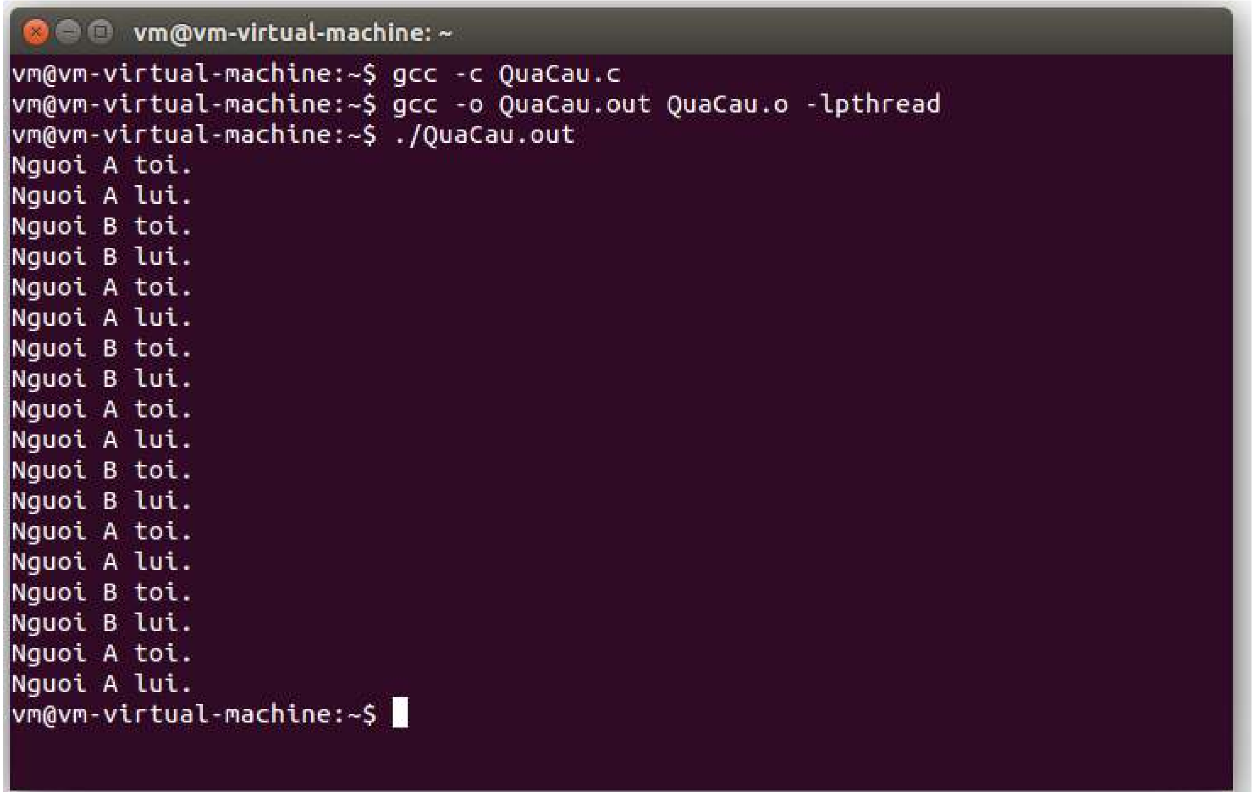
BÀI TẬP THÊM

Lab8.2

Bài 2:

```
#include <stdio.h>

#include <unistd.h>

#include <pthread.h>

#include <semaphore.h>



sem_t mutex1, mutex2;

void* W(void* arg)

{


          sem_wait(&mutex1);
```

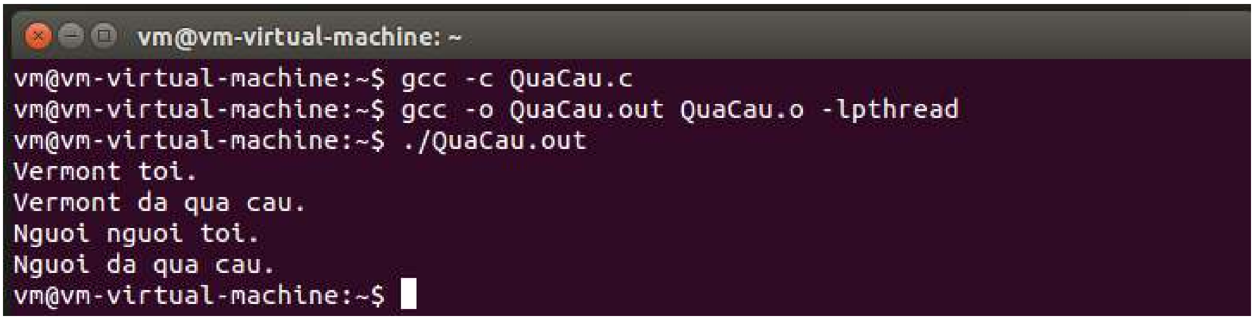
```
        printf("Nguoi A toi.\n");

        //critical section
        sleep(1);

        //Car is out
        sem_post(&mutex2);
        printf("Nguoi A lui.\n");
}

void* E(void* arg)
{

        sem_wait(&mutex2);
        printf("Nguoi B toi.\n");



        //Car is out
        sem_post(&mutex1);
        printf("Nguoi B lui.\n");
}

int main(void)
{
        pthread_t W1, W2, W3, W4, W5, E1, E2, E3, E4;
        sem_init(&mutex1, 0, 1);
        sem_init(&mutex2, 0, 0);

        pthread_create(&W1,NULL,W,NULL);
        pthread_create(&E1,NULL,E,NULL);
        pthread_create(&W2,NULL,W,NULL);
        pthread_create(&W3,NULL,W,NULL);
        pthread_create(&E2,NULL,E,NULL);
        pthread_create(&E3,NULL,E,NULL);
        pthread_create(&W4,NULL,W,NULL);
        pthread_create(&W5,NULL,W,NULL);
        pthread_create(&E4,NULL,E,NULL);

        pthread_join(W1,NULL);
        pthread_join(E1,NULL);
        pthread_join(W2,NULL);
        pthread_join(W3,NULL);
        pthread_join(E2,NULL);
        pthread_join(E3,NULL);
        pthread_join(W4,NULL);
        pthread_join(W5,NULL);

        sem_destroy(&mutex1);
        sem_destroy(&mutex2);

}
```



```
vm@vm-virtual-machine:~$ gcc -c QuaCau.c
vm@vm-virtual-machine:~$ gcc -o QuaCau.out QuaCau.o -lpthread
vm@vm-virtual-machine:~$ ./QuaCau.out
Nguoi A toi.
Nguoi A lui.
Nguoi B toi.
Nguoi B lui.
Nguoi A toi.
Nguoi A lui.
Nguoi B toi.
Nguoi B lui.
Nguoi A toi.
Nguoi A lui.
Nguoi B toi.
Nguoi B lui.
Nguoi A toi.
Nguoi A lui.
Nguoi B toi.
Nguoi B lui.
Nguoi A toi.
Nguoi A lui.
vm@vm-virtual-machine:~$
```

Bài 8.3

```
#include <stdio.h>

#include <unistd.h>

#include <pthread.h>

#include <semaphore.h>

sem_t mutex1, mutex2;

void* W(void* arg)

{

        sem_wait(&mutex1);

        printf("Vermont toi.\n");

        //critical section

        sleep(1);

        sem_post(&mutex2);

        printf("Vermont da qua cau.\n");
}

void* E(void* arg)
{
        sem_wait(&mutex2);
        printf("Nguoi nguoi toi.\n");
        sem_post(&mutex1);
        printf("Nguoi da qua cau.\n");
}

int main(void)
{
        pthread_t W1, W2, W3, W4, W5, E1, E2, E3, E4;
        sem_init(&mutex1, 0, 1);
        sem_init(&mutex2, 0, 0);
        pthread_create(&W1,NULL,W,NULL);
        pthread_create(&E1,NULL,E,NULL);
        pthread_join(W1,NULL);
        pthread_join(E1,NULL);
        sem_destroy(&mutex1);
        sem_destroy(&mutex2);

}
```



```
vm@vm-virtual-machine:~$ gcc -c QuaCau.c
vm@vm-virtual-machine:~$ gcc -o QuaCau.out QuaCau.o -lpthread
vm@vm-virtual-machine:~$ ./QuaCau.out
Vermont toi.
Vermont da qua cau.
Nguoi nguoi toi.
Nguoi da qua cau.
vm@vm-virtual-machine:~$ 
```