Status	Finished
Started	Monday, 24 March 2025, 9:23 PM
Completed	Monday, 24 March 2025, 9:29 PM
Duration	5 mins 46 secs
Grade	**100.00** out of 100.00

Question **1**
Correct
Mark 15.00 out of 15.00

Concurrency is possible in Uniprocessor systems.

Select one:

[✓ True] ✓

False

The correct answer is 'True'.

Question **2**
Correct
Mark 15.00 out of 15.00

A situation in which a runnable process is overlooked indefinitely by the scheduler is:

Select one:

a. Deadlock

b. Racing condition

[✓ c. Starvation] ✓

d. Mutual Exclusion

e. Livelock

The correct answer is:
Starvation

Information

Identify the number of critical section(s) and the shared resource(s) protected by the critical section in the following code.

```
#define NTHREADS 5
#define FAMILYNAME "CASTRO"

static pthread_cond_t empty = PTHREAD_COND_INITIALIZER;
static pthread_mutex_t bsem;
static int members=0;

void *access_house(void *family_void_ptr)
{
        char fam[20];
        char *ptr = (char *) family_void_ptr;
        strcpy(fam,ptr);
        pthread_mutex_lock(&bsem);
        std::cout << fam << " member arrives to the house" << std::endl;
        if (strcmp(fam,FAMILYNAME)!=0)
                pthread_cond_wait(&empty, &bsem);
        members++;
        std::cout << fam << " member inside the house" << std::endl;
        pthread_mutex_unlock(&bsem);
        sleep(5);
        pthread_mutex_lock(&bsem);
        std::cout << fam << " member leaving the house" << std::endl;
        members--;
        if (strcmp(fam,FAMILYNAME) == 0 && members == 0)
                pthread_cond_broadcast(&empty);
        pthread_mutex_unlock(&bsem);
        return NULL;
}

int main()
{

        pthread_t tid[NTHREADS];
        char family[NTHREADS][20];
        pthread_mutex_init(&bsem, NULL); // Initialize access to 1

        for(int i=0;i<NTHREADS;i++)
        {
                if(i%2 == 0)
                        strcpy((char *) &family[i],"RINCON");
                else
                        strcpy((char *)&family[i],"CASTRO");
                if(pthread_create(&tid[i], NULL, access_house,(void *)&family[i]))
                {
                        fprintf(stderr, "Error creating thread\n");
                        return 1;
                }

        }
        // Wait for the other threads to finish.
        for (int i = 0; i < NTHREADS; i++)
                pthread_join(tid[i], NULL);
        return 0;
}
```

Question **3**

Correct

Mark 10.00 out of 10.00

Shared Resource(s):

Select one or more:

- ☐ a. fam
- ☐ b. sleep(5)
- [✓ c. members] ✓
- ☐ d. family_void_ptr

The correct answer is: members

Question **4**

Correct

Mark 10.00 out of 10.00

Number of critical sections

Answer: 2 ✓

The correct answer is: 2

Information

Given the following matrices Q and A, and the available vector V, calculate the R vector and run the deadlock detection algorithm to determine the processes that are deadlocked.

$$Q = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Question **5**

Correct

Mark 10.00 out of 10.00

R = [$\boxed{3}$ ✓ $\boxed{4}$ ✓ $\boxed{2}$ ✓]

$\boxed{1}$ $\boxed{7}$ $\boxed{8}$ $\boxed{11}$ $\boxed{5}$

The correct answer is:

R = [[3] [4] [2]]

Question **6**

Correct

Mark 5.00 out of 5.00

Select the processes that are deadlocked

Select one or more:

- [ ] a. P5
- [ ] b. P2
- [ ] c. P1
- [x] d. P4 ✓
- [x] e. P3 ✓

The correct answers are:

P3

,

P4

Question **7**

Not answered

Not graded

Provide a file (JPEG, PDF, etc.) showing your work (step by step) while executing the Deadlock Detection algorithm.

Question **8**

Correct

Mark 35.00 out of 35.00

Complete the following C++ program to guarantee that only one person at a time will be in the house, alternating between a Rincon family member and a Castro family member (starting with a Rincon family member). Your program will receive from STDIN the number of people (npeople). The number of Rincon family members is ceil(npeople / 2) and the number of Castro family members is npeople - the number of Rincon family members.

For npeople = 5, the number of Rincon family members is 3 and the number of Castro family members is 2.

Expected output:

```
RINCON member inside the house
RINCON member leaving the house
CASTRO member inside the house
CASTRO member leaving the house
RINCON member inside the house
RINCON member leaving the house
CASTRO member inside the house
CASTRO member leaving the house
RINCON member inside the house
RINCON member leaving the house
```

NOTES

1. **You must create one thread per family member.**
2. **If you decide to use named POSIX semaphores, make sure that you select a unique name for your semaphores.**
3. **You can add as many variables as needed.**

For example:

Input	Result
5	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house

Answer: (penalty regime: 0 %)

Reset answer

```
#include <pthread.h>
#include <iostream>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <semaphore.h>
#include <fcntl.h>

static pthread_mutex_t bsem;
static pthread_cond_t rincon = PTHREAD_COND_INITIALIZER;
static pthread_cond_t castro = PTHREAD_COND_INITIALIZER;
static char turn[] = "RINCON";
static bool busy = false;

void *access_one_at_a_time(void *family_void_ptr)
```

```
{
    pthread_mutex_lock(&bsem);
    char fam[20];
    strcpy(fam,(char *) family_void_ptr);
    while (busy == true || strcmp(fam,turn)!=0)
    {
        if(strcmp(fam,"RINCON")==0)
            pthread_cond_wait(&rincon, &bsem);
        else
            pthread_cond_wait(&castro, &bsem);
    }
    busy = true;
    std::cout << fam << " member inside the house\n";
    pthread_mutex_unlock(&bsem);

    usleep(100);

    pthread_mutex_lock(&bsem);
    std::cout << fam << " member leaving the house\n";
    busy = false;
    if (strcmp(turn,"RINCON") == 0)
    {
        strcpy(turn,"CASTRO");
        pthread_cond_signal(&castro);
    }
    else
    {
        strcpy(turn,"RINCON");
        pthread_cond_signal(&rincon);
    }
    pthread_mutex_unlock(&bsem);

    return NULL;
}

int main()
{
```

	Input	Expected	Got	
✓	5	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house	✓

	Input	Expected	Got	
✓	8	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	✓
✓	2	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	✓
✓	10	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house	✓
✓	3	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house	RINCON member inside the house RINCON member leaving the house CASTRO member inside the house CASTRO member leaving the house RINCON member inside the house RINCON member leaving the house	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Cpp)

Correct

Marks for this submission: 35.00/35.00.