Musical Mayhem

There are 2 types of stages Acoustic and Electric and Some musicians and singers. This assigns stage to each of them with conditions given.

Running the Program

- Run gcc q3.c -lpthread;
- Run ./a.out

Input Format

- Enter the no. of Musicians, Acoustic and Electric Stages, Coordinators, Min and Max Duration, Waiting Time.
- Then, Enter each musician's information.

Functions

tshirt

```
sem_wait(&csem);
printf("\033[0;35m%s is collecting t-shirt\n",musician->name);
printf("\033[0m");
sleep(2);
printf("\033[0;33m%s left after Performing :,)\n",musician->name);
sem_post(&csem);
```

Give t-shirts to each musician and singer after performing. Only threads(musicians/singers) equal to number coordinators can get t-shirt at a same time. There is wait of 2 sec as it takes 2 secs to give a tshirt.

Acoustic

```
sleep(acoustic->arrival_time);
```

As particular musician takes this much time to arrive at Sarujana. So, sleep for that much time.

```
if((sem_timedwait(&asem,&ts) == -1) && errno == ETIMEDOUT)
{
    printf("\033[0;31m%s plays %c left due to impatience :,(\n",acoustic->name,acoustic->instrument);
    printf("\033[0m");
    return NULL;
}
```

Exit the function and print the required statement if particular musician waiting more than expected time.

```
for(ll i=0;i<(a+e);i++)
{
    pthread_mutex_lock(&(stage[i].mutex));
    if((stage[i].empty == 0) && (stage[i].type == 'a'))
    {
        stage[i].empty = 1;
        strcpy(stage[i].name, acoustic->name);
        stage[i].instrument = acoustic->instrument;
        stage[i].time = randomno(t1,t2);
        printf("\033[0;36m%s plays %c at acoustic stage number %lld for %lld sec\n",acoustic->name,acoustic->instrument,stage[i].id,stage[i].time);
        printf("\033[0m");
        pthread_mutex_unlock(&(stage[i].mutex));
        sleep(stage[i].time);
        if(stage[i].sings == 1)
        {
            sleep(2);
            sem_post(&stage[i].perform);
        }
        pthread_mutex_lock(&(stage[i].mutex));
        stage[i].empty = 0;
        pthread_mutex_unlock(&(stage[i].mutex));
        tshirt(acoustic);
        return NULL;
    }
    else
        pthread_mutex_unlock(&(stage[i].mutex));
}
```

Checks for empty acoustic stage, if empty, then assigned to acoustic musician to play for random time between t1 and t2. sleep stage for that particular time. Then checks if singer joins the musician or not, if yes, then sleep for 2 more sec as performance duration is increased by 2 secs. Make stage empty after performance and provide tshirt to both singer and musician and exit the function. If stage is not empty, then check for if musician is waiting for more than expected time or not till stage is provided or he/she left due to frustation.

Electric

```
sleep(acoustic->arrival_time);
```

As particular musician takes this much time to arrive at Sarujana. So, sleep for that much time.

```
if((sem_timedwait(&asem,&ts) == -1) && errno == ETIMEDOUT)
{
    printf("\033[0;31m%s plays %c left due to impatience :,(\n",electric->name,electric->instrument);
    printf("\033[0m");
    return NULL;
}
```

Exit the function and print the required statement if particular musician waiting more than expected time.

```
for(ll i=0;i<(a+e);i++)
{
    pthread_mutex_lock(&(stage[i].mutex));
    // printf("%lld\n",electric->id);
    if((stage[i].empty == 0) && (stage[i].type == 'e'))
    {
        stage[i].empty = 1;
        strcpy(stage[i].namelectric->name);
        stage[i].instrument = electric->instrument;
        stage[i].time = randomno(t1,t2);
        printf("\033[0;36m%s plays %c at electric stage number %lld for %lld sec\n",electric->name,electric->instrument,stage[i].id,stage[i].time);
        printf("\033[0m");
        pthread_mutex_unlock(&(stage[i].mutex));
        sleep(stage[i].time);
        if(stage[i].sings == 1)
        {
            sleep(2);
            sem_post(&stage[i].perform);
        }
        pthread_mutex_lock(&(stage[i].mutex));
        stage[i].empty = 0;
        pthread_mutex_unlock(&(stage[i].mutex));
        tshirt(electric);
        return NULL;
    }
    else
        pthread_mutex_unlock(&(stage[i].mutex));
}
```

Checks for empty electric stage, if empty, then assigned to electric musician to play for random time between t1 and t2. sleep stage for that particular time. Then checks if singer joins the musician or not, if yes, then sleep for 2 more sec as performance duration is increased by 2 secs. Make stage empty after performance and provide tshirt to both singer and musician and exit the function. If stage is not empty, then check for if musician is waiting for more than expected time or not till stage is provided or he/she left due to frustation.

Both

```
sleep(acoustic->arrival_time);
```

As particular musician takes this much time to arrive at Sarujana. So, sleep for that much time.

```
if((sem_timedwait(&asem,&ts) == -1) && errno == ETIMEDOUT)
{
    printf("\033[0;31m%s plays %c left due to impatience :,(\n",both->name,both->instrument);
    printf("\033[0m");
    return NULL;
}
```

Exit the function and print the required statement if particular musician waiting more than expected time.

```
for(ll i=0;i<(a+e);i++)
{
    pthread_mutex_lock(&(stage[i].mutex));
    if((stage[i].empty == 0))
    {
        stage[i].empty = 1;
        strcpy(stage[i].name, both->name);
        stage[i].instrument = both->instrument;
        stage[i].time = randomno(t1,t2);
        if(stage[i].type == 'e')
        {
            printf("\033[0;36m%s plays %c at electric stage number %lld for %lld
sec\n",both->name,both->instrument,stage[i].id,stage[i].time);
                        printf("\033[0m");
        }
        else
        {
            printf("\033[0;36m%s plays %c at acoustic stage number %lld for %lld
sec\n",both->name,both->instrument,stage[i].id,stage[i].time);
                        printf("\033[0m");
        }
        pthread_mutex_unlock(&(stage[i].mutex));
        sleep(stage[i].time);
        if(stage[i].sings == 1)
        {
            sleep(2);
            sem_post(&stage[i].perform);
        }
        pthread_mutex_lock(&(stage[i].mutex));
        stage[i].empty = 0;
        pthread_mutex_unlock(&(stage[i].mutex));
        tshirt(both);
        return NULL;
    }
    else
        pthread_mutex_unlock(&(stage[i].mutex));
}
```

Checks for empty stage(doen't matter which one, can perform on both), if empty, then assigned to musician to play for random time between t1 and t2. sleep stage for that particular time and checks for the type of stage too. Then checks if singer joins the musician or not, if yes, then sleep for 2 more sec as performance duration is increased by 2 secs. Make stage empty after performance and provide tshirt to both singer and musician and exit the function. If stage is not empty, then check for if musician is waiting for more than expected time or not till stage is provided or he/she left due to frustation.

Singer

```
sleep(acoustic->arrival_time);
```

As particular musician takes this much time to arrive at Sarujana. So, sleep for that much time.

```
if((sem_timedwait(&asem,&ts) == -1) && errno == ETIMEDOUT)
{
    printf("\033[0;31m%s plays %c left due to impatience :,(\n",singer->name,singer->instrument);
    printf("\033[0m");
    return NULL;
}
```

Exit the function and print the required statement if particular musician waiting more than expected time.

```
if((stage[i].sings == 0))
{
    stage[i].sings = 1;
    pthread_mutex_unlock(&(stage[i].mutex));
    strcpy(stage[i].singer_name, singer->name);
```

Checks if there is musician available or not. If stage is available, then checks for if it has musician on it or not.

```
if(stage[i].empty == 1)
{
    if(stage[i].type == 'e')
    {
        printf("\033[0;34m%s joined %s's performance at electric stage number %lld extended by 2 sec\n",singer->name,stage[i].name,stage[i].id);
        printf("\033[0m");
    }
    else
    {
        printf("\033[0;34m%s joined %s's performance at acoustic stage number %lld extended by 2 sec\n",singer->name,stage[i].name,stage[i].id);
        printf("\033[0m");
    }
sem_wait(&stage[i].perform);
pthread_mutex_lock(&(stage[i].mutex));
stage[i].sings=0;
pthread_mutex_unlock(&(stage[i].mutex));
tshirt(singer);
}
```

If musician is present then, type of stage is checked and accordingly printed. As, musician left the stage, Singer simultaneously left too. Stage is made empty/available.Then , both go to recieve t-shirt.

```
else
{
    stage[i].empty = 1;
    stage[i].tirandomno(t1,t2);
    pthread_mutex_unlock(&(stage[i].mutex));
    if(stage[i].type == 'e')
    {
        printf("\033[0;36m%s starts solo performance at electric stage number %lld for %lld sec\n",singer->name,stage[i].id,stage[i].time);
        printf("\033[0m");
    }
    else
    {
        printf("\033[0;36m%s starts solo performance at acoustic stage number %lld for %lld sec\n",singer->name,stage[i].id,stage[i].time);
        printf("\033[0m");
    }
    sleep(stage[i].time);
    pthread_mutex_lock(&(stage[i].mutex));
    stage[i].sings=0;
    stage[i].empty=0;
    pthread_mutex_unlock(&(stage[i].mutex));
    tshirt(singer);
}
```

If there is no musician, singer gives solo performance and type of stage is checked. As, musician left the stage, Singer simultaneously left too. Stage is made empty/available. Then , both go to recieve t-shirt.

> *At the end Finished is printed.*

Main

- Take input
- Create threads(Musicians) and join them.
- Initiate Mutexes and Semaphores.
- Destroys Semaphores and Mutexes.

Bonus

- Both the bonuses are implemented.
- Tshirt is give to singers too.
- Each time stage is taken care of(1 to a stages are acoustic and remaining ones are electric).