REPORT OF QUESTION 3

The code uses five semaphores to implement the question and their function is as follows

- Acoustic_sem - initialized to the number of the acoustic stages and is used to assign a musician to the acoustic stage, for example, the musician who plays the violin wait on this semaphore until the maximum waiting time
- Electric_sem - initialized to the number of electric stages and is used to assign a musician to an electric stage, for example, the musician who plays bass will wait on this semaphore until the maximum waiting time
- Total_sem - initialized to the total number of stages and is used to check for a free musical stage for a musician to play which can play on either stage. Musicians who can play on both the stages wait on this semaphore for the maximum waiting time
- Singer_sem - initialized to the total number of stages and is used to check for a position for a singer as the singers can take at most number of stages positions and singers wait on this semaphore for the maximum waiting time
- Coordinators - initialized to the total number of coordinators and is waited by singers and musicians to wait on this semaphore to collect their t-shirt

ASSUMPTIONS

- If a singer joins the performance of a musician then they both go to collect the t-shirt together
- The singer also goes after the maximum waiting time due to impatience
- Electric Stages are numbered from 1 and Acoustic stages are numbered after the electric stages

DATA STRUCTURES

- Structure Performer which stores
    - Name of the performer
    - Code like ‘s’ or ‘g’ etc.

    - Arrival_time: time after which the performer reaches
    - Partner - address to the partner who is playing on stage, basically for musicians
    - Thread Id
- Stage - An array of Stages which stores the structure information pointer as the information of the performer who is playing on the stage

FUNCTIONS

- Swap() and ShuffleArray() - to shuffle the index of the available stages to randomize the stage selection process for a performer to choose from
- Music() - Function called by the musician threads and all implementation done here for the musicians
    - When the thread arrives it sleeps for the arrival time of the performer
    - Then the flags are set if it can play on both stages or only one of them
    - Then the flags are set for which stages the performer can play on and sem_timedwait() is used accordingly with the required semaphore for the maximum waiting time
    - If the time is spent then the functions prints and returns
    - If the wait is executed before the waiting time then variables are set like
        - Duration of the performance
        - An array of the stage numbers where the performer can perform is set and shuffled for randomization
    - Then the array is traversed and the loop checks if the current stage has a performer playing on and continues if it has
    - If it doesn't then the array Stage is set with this musician on the index of the stage number
    - If the performer is only a single stage player then it would have already waited on the stage-specific semaphore and we just

need to decrement the total_sem semaphore to indicate a stage is booked
    - Then it sleeps for the duration of the performance and sleeps for extra 2 seconds if it has a partner in the performance and then resets all the variables and frees the stage and post the required semaphores
    - Then the musician waits on the coordinator semaphore and sleeps for 2 seconds after that to collect the t-shirt and prints the message appropriately
- Sing()
    - Similar starting like the music() function sleeps for the arrival time, waits on the singer_sem for maximum waiting time and sets the required variables.
    - If the stage is occupied by a singer or already there two performers then the loop continues
    - Then if the singer can join it joins and sets the partner of the musician with its address and returns the function and the semaphore is posted by the singer when it finishes the performance
    - Otherwise, it tries waiting on the semaphore for the type of stage and if it is free it is assigned to the stage and semaphores are posted accordingly.
    - Then if the singer didn't join the performance then they collect the t-shirt by waiting on the coordinators' semaphore and sleeps for 2 seconds to collect the t-shirt and return from the function
- main()
    - Take input and initialize all the semaphores
    - Allocate memory to required variables and store the information about the performers in an array
    - Use that array afterwards to create threads for each performer and wait on each thread after making them