

This project is completed by:

Dilara Deveci - 0068182

Fulya Akın - 0064220

The code is compiled with

`g++ code.cpp -lpthread -o code`

It is run with

`./code -n 4 -p 0.75 -q 5 -t 3 -b 0.05`

The zip file includes:

code.cpp,

README.pdf

Brief explanation of our implementations:

## **Part 1**

### **Variables**

- We have two different types of threads which are created with `moderator_func` and `commentator_func`.
- Three condition variables and one mutex are used to synchronize the threads.
- The mutex named `mic` works like a real life microphone and the one who wants to speak should lock the mic while speaking.
- The condition variable named `questionAsked` signals that the moderator asked a question, now the commentators who are waiting for this signal can start to speak.
- The condition variable named `answer_request` signals that the commentators who have been added to the queue want to speak and need the permission of the moderator to speak.
- The condition variable named `you_may_speak` signals that commentators now have the permission of the moderator and can continue to execute.
- The condition variable named `questionAnswered` signals that the commentators answered the question. If there are questions left, the moderator can ask the next question. If there is no question, the session can be terminated.

### **Execution**

- We take the number of commentators, speak time for commentators, number of questions, probability of answer and the probability of the breaking news with the command line arguments.

- Firstly, the `commentator_func` waits for the `questionAsked` signal and the `moderator_func` asks the first question. The `commentator_func` starts adding the commentators to the queue. Then, the commentators who have been added to the queue signal the `answer_request` and wait for the `you_may_speak` signal. The moderator who has been waiting for the `answer_request` sends the `you_may_speak` if the queue is not empty. After all commentators in the queue have spoken for a random amount of time, the `commentator_func` signals the `questionAnswered` and starts again to wait for the `questionAsked`. Until completing all the questions, the process repeats itself.
- It print following logs for each commentator and for each question:  
 "[0:0] The moderator asks question #"  
 "[0:0] Commentator # generates answer,position in queue: #"  
 "[0:0] Commentator #'s turn to speak for 2.50592 second"  
 "[0:2.50592] Commentator #2 finished speaking."  
 "End of the session".

## Notes

- If there is no question or no commentator in the beginning, the session is ended.

## Part 2

- The condition variable named `news` is used by the moderator to signal to the `breaking_func` that there is breaking news.
- The condition variable named `finishedNews` signals that the breaking news ended from the `breaking_func` to the `moderator_func`.
- When the `breaking_func` has taken the signal `news`, it prints the log "Breaking news!", then it sleeps for 5 seconds. Then, it prints the log "Breaking news ends". It unlocks the mic mutex.

## Notes

- We could not find a way to cut off the execution of a commentator while it's sleeping therefore the `moderator_func` first checks whether the breaking variable which is randomly generated is smaller than the  $b*100$ , then if it is smaller, it signals to the `breaking_func` to execute the breaking news. If it is bigger than  $b*100$ , it signals to the commentator that the microphone is theirs.

### Part 3

Example logs of one session with  $N=3$ ,  $q=4$ ,  $p=1$ ,  $t=3$ ,  $b=0.05$

```

1
dilara@dilara:~$ ./code -N 4 -p 0.75 -q 5 -t 3 -b 0.05
[0:0] Moderator asks question 1
[0:0] Commentator #2 generates answer,position in queue: 0
[0:0] Commentator #3 generates answer,position in queue: 1
[0:0] Commentator #3 generates answer,position in queue: 2
[0:0] Commentator #3's turn to speak for 1.88661 second
[0:1.88661] Commentator #3 finished speaking.
[0:1.88661] Commentator #4 generates answer,position in queue: 3
[0:1.88661] Commentator #4's turn to speak for 2.74859 second
[0:4.6352] Commentator #4 finished speaking.
[0:4.6352] Moderator asks question 2
[0:4.6352] Commentator #1 generates answer,position in queue: 0
[0:4.6352] Commentator #1's turn to speak for 1.82091 second
[0:6.4561] Commentator #1 finished speaking.
[0:6.4561] Commentator #1 generates answer,position in queue: 1
[0:6.4561] Commentator #1's turn to speak for 0.411695 second
[0:6.8678] Commentator #1 finished speaking.
[0:6.8678] Commentator #3 generates answer,position in queue: 2
[0:6.8678] Commentator #3's turn to speak for 1.20283 second
[0:8.07063] Commentator #3 finished speaking.
[0:8.07063] Commentator #4 generates answer,position in queue: 3
[0:8.07063] Commentator #4's turn to speak for 2.99677 second
[0:11.0674] Commentator #4 finished speaking.
[0:11.0674] Breaking news!
[0:16.0674] Breaking news ends
[0:16.0674] Commentator #2's turn to speak for 1.83792 second
[0:17.9053] Commentator #2 finished speaking.
[0:17.9053] Commentator #3's turn to speak for 1.91266 second
[0:19.818] Commentator #3 finished speaking.
[0:19.818] Moderator asks question 3
[0:19.818] Commentator #4 generates answer,position in queue: 0
[0:19.818] Commentator #4's turn to speak for 2.31407 second
[0:22.1321] Commentator #4 finished speaking.
[0:22.1321] Commentator #2 generates answer,position in queue: 1
[0:22.1321] Commentator #2's turn to speak for 1.20069 second
[0:23.3327] Commentator #2 finished speaking.
[0:23.3327] Commentator #1 generates answer,position in queue: 2
[0:23.3327] Commentator #1's turn to speak for 1.05738 second
[0:24.3901] Commentator #1 finished speaking.
[0:24.3901] Commentator #4 generates answer,position in queue: 3
[0:24.3901] Commentator #4's turn to speak for 2.84798 second
[0:27.2381] Commentator #4 finished speaking.
[0:27.2381] Moderator asks question 4
[0:27.2381] Commentator #1 generates answer,position in queue: 0
[0:27.2381] Commentator #1's turn to speak for 2.6707 second
[0:29.9088] Commentator #1 finished speaking.
[0:29.9088] Commentator #3 generates answer,position in queue: 1
[0:29.9088] Commentator #3's turn to speak for 0.0600691 second
[0:29.9689] Commentator #3 finished speaking.
[0:29.9689] Commentator #2 generates answer,position in queue: 2
[0:29.9689] Commentator #2's turn to speak for 2.70662 second
[0:32.6755] Commentator #2 finished speaking.
[0:32.6755] Commentator #1 generates answer,position in queue: 3
[0:32.6755] Commentator #1's turn to speak for 1.61928 second
[0:34.2948] Commentator #1 finished speaking.
[0:34.2948] Moderator asks question 5

```

```
[0:34.2948] Commentator #1 finished speaking.  
[0:34.2948] Moderator asks question 5  
[0:34.2948] Commentator #2 generates answer,position in queue: 0  
[0:34.2948] Commentator #2's turn to speak for 1.59482 second  
[0:35.8896] Commentator #2 finished speaking.  
[0:35.8896] Commentator #3 generates answer,position in queue: 1  
[0:35.8896] Commentator #3's turn to speak for 2.16286 second  
[0:38.0524] Commentator #3 finished speaking.  
[0:38.0524] Commentator #1 generates answer,position in queue: 2  
[0:38.0524] Commentator #1's turn to speak for 1.91994 second  
[0:39.9724] Commentator #1 finished speaking.  
[0:39.9724] Commentator #4 generates answer,position in queue: 3  
[0:39.9724] Commentator #4's turn to speak for 0.497923 second  
[0:40.4703] Commentator #4 finished speaking.  
End of the session  
dilara@dilara:~$
```