KRSSG SOFTWARE TASK ROUND

Task 1

Unlikely Redemption: Prisoner Heroes

During the devastating aftermath of Hurricane Katrina, a group of prisoners who were part of an emergency response program emerged as unsung heroes. These individuals, despite their incarcerated status, volunteered to assist in rescue and relief efforts. As a token of appreciation for their extraordinary contributions, the government decided to award these prisoners for their invaluable assistance in saving lives and providing much-needed aid during this challenging time.

The government entrusted you with the responsibility of ensuring that no undeserving prisoner would be rewarded, and you devised a meticulous game plan to guarantee just that. The prisoners would need to follow an escape plan, and their rewards would be distributed in accordance with the order of their successful escape.

Game Plan:

- You (server) have to generate two random numbers **L and R.** Make sure that the difference between these two numbers is at least 10^4 and at most 10^5. You then have to generate a random number in the range **[L,R].** Let's call this number **X**. This process is to be performed only once at the start of the game.
- Each of the prisoners (clients) will receive the numbers **L** and **R** from the server at the start of the game.
- At each timestep of the game, the prisoners will guess a random number (Let's call this value **Y**. Note that all prisoners will be sending random numbers so this value **Y** need not be the same for all the prisoners) in the range and send that value to the server. The server analyzes the value sent to it and three cases may arise:
    1. Y>X. In this case the server sends a message to the client stating that the value is too high.
    2. Y<X. In this case the server sends a message to the client stating that the value is too low.
    3. Y=X. In this case the server sends a message to the client stating that the client has guessed the correct value.
- Based on the message sent by the server the clients will update their search ranges for the random number.
- After a certain finite number of timesteps, all prisoners would have guessed the number **X** correctly and would escape the prison with the reward.
- Whenever a prisoner escapes, print "<Prisoner Name> escaped" on the terminal of every other prisoner (client). Also, close the client connection once that prisoner(client) escapes.

- After all the prisoners escape, print the order of escape in the server terminal for fair distribution of the rewards! **:)**

Program Files:

You have to make two scripts in C++.

1. **Game Moderator (Server):** Generates the initial search range **[L,R]** and the random number **X** and assigns ID/Name to each prisoner. At the beginning of the game, sends the two numbers "**L**" and "**R**" to the prisoners. At each timestep, accepts the guess (**Y**) from each prisoner (client), and sends an appropriate message to each prisoner(client) as mentioned above. Also, prints which prisoner has escaped on all the clients' terminals. Lastly, prints the order in which prisoners have escaped in its own terminal.
2. **Prisoners (Clients):** Sends a random number in the current search range to the server and depending on the server message updates the search range for future timesteps.

   Note: All the prisoners must run the same script.

Constraints:

- Number of prisoners = 4
- 10^4 <= R-L <= 10^5
- L<=X<=R

Bonus:

1. Make a separate client script which asks for the number of prisoners to escape prison.

2. Connect the same number of clients to the server through the client script.

3. Implement a **Shared Pool** feature in the client and server scripts. This feature is something that allows the clients to indirectly communicate with each other through the server. At the start of the game, every client has to select whether they want to be a part of the Shared pool or not. If they are not a part of the shared pool, there is not much of a difference than the client previously implemented. If the client wants to be a part of shared pool, you need to implement the following functionalities:

- The information about their guess is shared to the pool.
- The information of the pool is received by the client through the server after each time step.
- You need to implement an effective way of sharing the information of the pool to each client. You may want to discover a way of embedding all the information in less space.
- The client uses the information to modify their guess ranges so as to land a better guess.

Language: C++

Suggested Reading : Socket Programming , Multithreading