OVERVIEW

Design of code:

The complete program is split into 3 parts.

- **postoffice.hpp** – contains implementation of the priority queue,
- **whatsapp.hpp** – imports the postoffice.hpp file and implements the thread pool and related function,
- **main.cpp** – imports the threadpool class and creates an object of it, passing the required message information as arguments.

Here's whats happening basically.

All the threads have some messages to send to other threads. They have these messages stored in a vector of their own. They iterate through their vectors, sending messages to the queue on each iteration. Now, on each iteration when they get access to the message_queue, before sending in a message they also check if theres a message for them on top or not. This way the the program becomes faster and more efficient.
After a thread is done sending all the messages they had to send, it waits. It waits untill theres a message for it in the message_queue. It does so by iterating a while loop which stops when there are no other messages for that thread. Other threads might still be conversing with each other but if they don't have anything to say to this thread, its gonna stop running the loop and finish its execution.

Here's an informal explanation of the algorithm.

Imagine all the threads are people. Now for some reason they cannot talk with each other so they decide to commincate using postcards. They all have written a bunch of postcards to send to other people. On each postcard is written who it's from and who it's intended for. They also have a magic box which they share and only one person can use it at a time. Whenever you put in a bunch of postcards in that magic box it will automatically float up the one which is most important.

So now, one-by-one they come and drop in their postcard in the box. Since there are a bunch of people and only one box, it would be unfair to keep the box occupied by one person for too long. What if one has 100 postcards to drop in while the person next in line only has 1? Everyone should get a fair chance. Therefore, after someone drops in one card of

their own they leave the box unoccupied for some time (a few miliseconds). Whoever can come first now gets to use it.

Now, suppose one goes in to drop their card and while dropping it they can see that the card on the top is meant for them. It'd be a waste to time to wait till he/she has dropped all their cards off to get it. It might not even be at the top then. Frustrating no? That's why if they see a card meant for them at the top they'll pick it up and carry on.

What happens after someone has dropped in all their cards off? Well, through some magical way they know how many postcards are meant for them. So they wait till they can get all the postcards meant for them. Whenever someone puts in their card or takes out a card, they shout 'New Cards!! New Cards!! Come check for your postcards!!' since the magic box has now been updated (They cannot call a specific person, either all of them or one random person). Everybody comes rushing in to see if they've got a message or not. If they do, they take it out and shout the same thing. If not, they go back to waiting.

After one has both, got all the cards meant for them as well as dropped in all the cards they wanted to send, they don't need to use the magic box anymore. They are done for the day. So they just sit and rest till all the other people are done, and then one-by-one they all go back home.