Lab 13

1. ***Hot Potato 2: Electric Boogaloo**:* Write a C program that receives a number N as a command-line argument. The main process generates a random integer between 1000 and 10000 (we'll call this variable POTATO) and creates N *player* threads and one (1) *monitor* thread. Each thread will wait until all other threads have started.

   Each player thread gets a unique identifier from the main process starting at 1. The N player threads will execute an infinite loop in which they try to subtract a random value between 10 and 100 from the POTATO and then sleep for a random amount of time between 10 and 20 milliseconds. The first thread that causes the POTATO to have a negative value prints a message that announces this alongside its given identifier, breaks the loop, and terminates. If a player thread detects that the POTATO is negative, it signals the monitor thread and waits for the POTATO to be reset.

   The monitor thread keeps track of how many player threads are still alive and waits for the POTATO to become negative. When the POTATO becomes negative, it resets it with a random integer between 1000 and 10000 and decreases the number of alive threads.
   This cycle repeats until there is only one player thread left, which is declared as the winner.
   (You can expand on the solution for the first hot potato:
   https://www.cs.ubbcluj.ro/~horea.muresan/os/sol-c/36.c)

2. Write a C program that receives 3 command-line arguments: Src, Dest, and N. Src and Dest are filenames, and Src must already exist, Dest may already exist, but the program will create it if it does not exist. N is an integer. The program will copy file Src into file Dest using N threads, where each thread will repeatedly read chunks of 20 bytes from Src and write them to Dest. Ensure that the Dest file is a correct copy of the Src file.