HAMZA UMER

BSCS 4 B

OS LAB 10

**Q1: filename: race\_threads.c  
Description:**

Two threads are made. In both the threads loops are there with pthreadexit

Which means both threads will run simultaneously, first incrementing, then decreasing. BUT WE DO NOT know how many times the individual thread will run before ending the individual process. After loop ends. It will give the balance value which will be different each time we run it. THIS CONDITION IS CALLED RACE CONDITION

**Q2: Filename: race2.c  
Description:**

Same as Q1. 2 Threads are created. Which take arguments from main from user entered at runtime. Then it calculates number of characters in the files and calculate total number of characters and return to main program. After the thread has counted number of characters it is destroyed

**Q3: Filename: t2.c  
Description:**

SAME AS Q1. Two threads are made. In both the threads, infinite loops are there with pthreadexit

Which means both threads will run simultaneously, BUT WE DO NOT know how many times the individual thread will run before ending the individual process. Value which will be different each time we run it. THIS CONDITION IS CALLED RACE CONDITION

**Q4: Filename t3.c  
Description:**

It takes arguments from user at runtime and creates two threads. After that it prints first thread to the size of first argument and same for second thread. So if we type ./q4 4 5. It prints XXXXOOOOO

**Q5: Filename: t4.c  
Description:**

This creates two threads and gives a size of 1000 to loop to each. First the first thread prints X 1000 number of times then O 800 times. Then main prints bye bye and destroys thread.

**Q6: Filename: id\_threads1.c  
Description:**

This program creates thread. First it prints the PID and TID of main. Then it goes in thread and prints its PID and TID. 2 threads are created and both show their ID’s. SINCE THERE IS NO “PTHREAD\_JOIN”, the program will never end and be in a deadlock state.

**Q7: Filename:   
Description:**

This program shows PID, TID and pthread\_self() of all 3 threads