Exercise 1: Create a Hello World program using threads

Exercise 2: Modify exercise 2, to print the thread id of the threads along with the Hello World message

Exercise 3: Write a program hellomany.c that will create a number N of threads specified in the command line, each of which prints out a hello message and its own thread ID

Exercise 4: Write a program using threads, where the main thread increments the value and the child thread decrements the same value.

Exercise 5 : Given two character strings s1 and s2, use C and pthread to write a parallel program to find out the number of substrings, in string s1, that are exactly the same as string s2. The strings are ended with '\0'. For example, suppose number_substring(s1, s2) implements the function, then number_substring("abcdab", "ab") = 2, number_substring ("aaa", "a") = 3, number_substring("abac", "bc") = 0. Suppose the size of s1 and s2 are n1 and n 2, respectively, and p threads are used, we assume that n1 mod p = 0, and n2 < n1/p.

Strings s1 and s2 are stored in a file named "strings.txt". String s1 is evenly partitioned for p threads to concurrently search for matching with string s2. After a thread finishes its work and obtains the number of local matching, this local number is added into a global variable showing the total number of matched substrings in string s1. Finally this total number is printed out. The format of the strings.txt is like this (the first string is s1 and the second one is s2):

abcassghbcaj
bca

In the program, use #define to specify number of threads to be created. For example,
#define NUM_THREADS 5

Exercise 6 : Given an array of integers, use C and pthread to write a parallel program to find out the sum of the array and the second maximum. Assume the entire array is store d initially in one location and is distributed to the different threads for parallel processing.