Explanation for Q1

Part 1:

Output Observed:

We observe that in the **parent process** the value of x before the while loop is 10 and it increments by 1 in each iteration of the while loop till its value becomes 100, and the final value at the end of the loop is 100.

We observe that in the **child process** the value of x before the while loop is 10 and it decrements by 1 in each iteration of the while loop till its value becomes -90, and the final value at the end of the loop is -90.

Explanation of Output:

After forking a different physical address space is created for the child process and the values of each variable present in the parent's physical address space at the time of forking is copied by value in the child's physical address space.

Due to this fact the variable x in the child and parent process are located at different positions in the memory and therefore a change in x in the parent process does not change x in the child process.

Part 2:

Output Observed:

The value of x before the start of **parent** thread loop 10. It increments by 1 in each loop till it becomes 100, and the final value after the loop is 100.

The value of x before the start of the **child** thread loop is 100. It decrements by 1 in each loop till it becomes -90, and the final value after the loop is -90.

Order of values inside and before the loop changes every time the program is run.

Explanation of Output:

Unlike fork() when a thread is created no new physical address space is created for the child thread. Hence, the parent and child thread share the same physical address space. When means modification of the variable in one thread reflects the change in the other.

Due to this fact when the program is run **race condition** occurs between the child and parent thread as they are changing and accessing the variable x which is located at the same physical address for both the threads.
Outputting the variable x in the while loop in both the threads, allows us to see the context switching or scheduling happening with both the threads.