Case1 : fork

In fork when a child process is created a copy of the entire memory space is also made thus the memory space of the child process and parent process are different, thus when the global variable being accessed in both the processes are different as their virtual addresses are different thus we can see that first the parent process increments the value of the variable to 100, but this doesn’t affect the value of the variable in the memory space of the child process and thus the child process decrements the value of the global variable from 10 to -90.

Case2: pthread\_create

In pthread\_create the child thread and the parent thread share the same memory space as no separate copy of the memory space is created due to this they both are referencing to the same value in the memory space, thus this means that even if one thread updates the value of the global variable the other thread would be able to see the updated value only due to the shared memory space.

Thus we observe that first the child thread updates the value of the global variable to -90 and then the parent thread changes the values from -90 only to 100.