Q1.
In question 1, I have created two files named q1_1.c and q2_2.c. In the q1_1.c file, I have used a fork to create a child and parent process, I have not used the wait to make my program wait for the child process to complete.
In the q1_2.c file, I have made threads to complete the task that was specified in the question.

The values of first and second file is that when we run first program, the initial value of x in the child program is 10 and the final value is -90. In the parent process, the initial value is 10 and the final value is 100.
In the second file, the initial value of x in the child program is 100 and the final value is -90. In the parent process, the initial value is 10 and the final value is 100.
Here the values of child and parent execution can be different according to the scheduling of the child and parent process.

The difference in the values in the first and second file execution is that in the process, the parent and child both get the initial value of x to be 10 because here the child and parent have their separate copies of x, due to which it doesn't matter how the scheduler schedules the event, both are going to get the value of x to be 10 and would thus give -90 and 100 as output.
In the case of threads, the value of x is global, both parent and child process needs to use the same value of x due to which the thread that run first changes the value of x and the final value of that thread would become the initial value of the next thread that would be executed. Here which thread would be executed first depends on the scheduler.