Write-Up

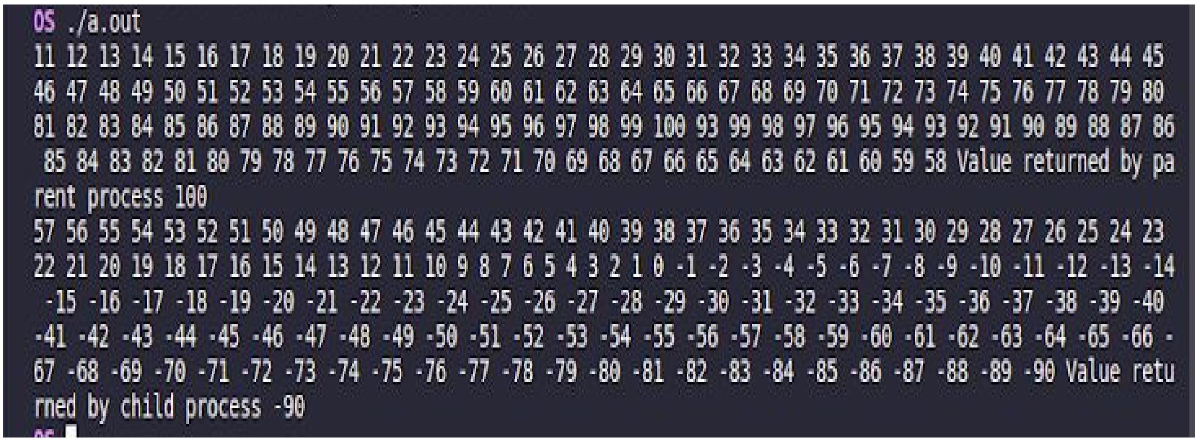
There is a difference between the values printed by the child & parent processes compared to the thread of child and parent.

In Q1_Part1: using fork()

Parent and child Process contain their own variables in their own address spaces. Its mechanism is not to share a memory space. The child process and the parent process work separately and cannot touch the process—variables of or each other of another method. Hence, the child process has its own value of the global variable of x and is decrementing its own copy of the global variable. Same for the parent process. Both the child and the parent process are working on their own value of the global variable x.

In Q1_Part2: using pthread_create()

Thread has a distinctive structure and shares its memory space. Thus, threads partially copy all the data of the parent and work on the same variable x, like that of the parent thread. The child and parent thread production Run together and increase from 10 to 100 for the 1st and then decrease the same value of x to a negative ninety. Threads share the same memory and may impact some threads' variables. As both the parent and child thread are accessing the same value of the global variable x and are both running concurrently, any of the threads change the value of x at any time until its operation is complete. This can be clearly observed here in the output.



```
OS ./a.out
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 93 99 98 97 96 95 94 93 92 91 90 89 88 87 86
 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 Value returned by pa
rent process 100
57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23
22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14
 -15 -16 -17 -18 -19 -20 -21 -22 -23 -24 -25 -26 -27 -28 -29 -30 -31 -32 -33 -34 -35 -36 -37 -38 -39 -40
-41 -42 -43 -44 -45 -46 -47 -48 -49 -50 -51 -52 -53 -54 -55 -56 -57 -58 -59 -60 -61 -62 -63 -64 -65 -66 -
67 -68 -69 -70 -71 -72 -73 -74 -75 -76 -77 -78 -79 -80 -81 -82 -83 -84 -85 -86 -87 -88 -89 -90 Value retu
rned by child process -90
```