(1) Parent sets up a string which is read by child, reversed there and read back the parent

(2) Parent sets up string 1 and child sets up string 2. String 2 concatenated to string 1 at parent end and then read back at the child end.

(3) Substring generation at child end of a string setup at parent process end.

(4) String reversal and palindrome check using pipes / shared memory.

(5) Armstrong number generation within a range. The digit extraction, cubing can be responsibility of child while the checking for sum == no can happen in child and the output list in the child.

(6) Ascending Order sort within Parent and Descending order sort (or vice versa) within the child process of an input array. (u can view as two different outputs –first entire array is ascending order sorted in op and then the second part descending order output).

(7) Implement a multiprocessing version of binary search where the parent searches for the key in the first half and subsequent splits while the child searches in the other half of the array. By default u can implement a search for the first occurrence and later extend to support multiple occurrence (duplicated elements search as well).

(8) Read upon efficient ways of parallelizing the generation of Fibonacci series and apply the logic in a parent child relationship to contribute a faster version of fib series generation.

(9) Longest Common Subsequence (Extra-Credits)