(1) Test Drive all the examples discussed so far in the class for the usage of wait, exec call variants.

For the following questions you are free to decide the responsibility of parent / child processes. As mentioned in the class u r allowed to use vfork / file based approach to avoid the data sharing issues which we will later address using pipes in later classes to follow.

(2) Odd and Even series generation for n terms using Parent Child relationship (say odd is the duty of the parent and even series as that of child)

(b) given a series of n numbers (u can assume natural numbers till n) generate the sum of odd terms in the parent and the sum of even terms in the child process.

(3) 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.

(4) Fibonacci Series AND Prime parent child relationship (say parent does fib Number generation using series and child does prime series)

(5) 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 asc order sorted in op and then the second part desc order output)

(6) Given an input array use parent child relationship to sort the first half of array in ascending order and the trailing half in descending order (parent / child is ur choice)

(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) * Non Mandatory [extra credits]

Read upon efficient ways of parallelizing the generation of Fibonacci series and apply the logic in a parent child relationship to contributes a faster version of fib series generation as opposed to sequential logic in (4)