1. The program scans integers from a file and stores them in a vector. The integers are cross-verified with the command-line argument and checked for out-of-bound errors or invalid data.
2. When everything is done, an infinite loop is used to accept a value from the user and search for it in the array and repeat the process until the user types “quit”.
3. After accepting a proper integer, the program uses shmget() to create a shared memory segment, and then attaches it with shmat().
4. Using copy() the vector is copied to the shared memory.
5. Then a for loop is used to create a number of child processes requested by the user. Each process only gets a portion of the array to scan for the value using the method count().
6. The shared memory is accesses by the child processes using a pointer pointing to their respective portion of the memory.
7. At the end of every child process, shmdt() is used to detach it from the shared memory segment.
8. After that exit() is used exit the process with the result of the computation in count()
9. When all the work is done another for loop is used for the parent process to wait for all the child processes using wait() and get the exit status(return value from count()) using WEXITSTATUS().
10. All the values returned by WEXITSTATUS() are added and then printed on the screen including what every process returned.