Report:

1. If these signals can be handled at the operating system level, their handler functions can be altered by the programmer. This way the programmer can define how the program will behave when these signals are received especially if they do not desire their program killed or stopped in conventional manner.

2. Pause() will suspend program execution until a signal is received that has a handler function that can be executed. Putting pause() inside an infinite while loop will prevent the program from wasting CPU cycles.

3. Other signals are masked to block them and prevent those signals from interrupting critical sections of code carried out by the current signal's handler function.

4. SIGALRM is a signal that interrupts the program and calls its handler function. We keep this unmasked in order to activate an interrupt upon time out. Do not mask or this signal is blocked and cannot interrupt.