**Problem 1 Solution Description:**

The working files in my code are as follows:

1. **main.c:** The file main.c is the parent program that creates 3 child processes using the fork() system call.
    - Firstly, the main forks process S1, which defines the SIGTERM Handler using the function sigaction(), and inside S1's SIGTERM handler which reads the message and prints it. (I have used the concept of shared memory to pass the strings from E1 and E2 to main  which i will explain later)
    - Secondly, the main process forks two more processes SR and ST which further execute the programs E1 and E2 respectively using execv() system call and have passed the pid (process id) of S1 in the arguments.
      Main has sleep(20) at the end after which it kills all it child processes using SIGKILL syscall
2. **E1.c:** The file E1.c contains a SIGALRM Handler which sends a random number to S1 (using shared memory) and sends a SIGTERM signal to S1 using the sigqueue() system call.
   Note: To read the random number from the system I have used the rdrand instruction with asm volatile keyword to so that it is not blocked by the compiler optimization.
   E1.c also contains the syscall setitimer() to send the SIGALRM signal at every interval of 1 second.
3. **E2.c:** The file E2.c contains a SIGALRM Handler which sends a signal to E2's SIGTERM Handler which further sends a string containing date and time  SIGTERM signal to S1 using the sigqueue() system call similar to the E1.c.
   Note: To read the timestamp counter from cpu I have used the rdtsc instruction with asm volatile keyword to so that it is not blocked by the compiler optimization.
   Further it converts the value obtained by rdtsc to human-readable format using the function asctime() present in <time.h> header file.

**Shared Memory:**

For passing the string from E1 and E2 to main I have used the concept of shared memory where the database file is used for shared memory.

For shared memory functions used are:

- *ftok()*
- *shmget()*
- *shmat()*
- *shmdt()*
- *shmctl()*

These functions are used to attach and detach memory blocks from the program so that each file can have shared memory.

E1 and E2 write in the database and main reads from the database and further prints. These functions help in maintaining successful IPC .