Assignment 6 (CST-102)

1. Compllete the following C code where child process calls WAIT_PARENT function to wait for parent and continues only after receiving SIGUSR1 signal from parent. Parent executes TELL_CHILD function to send SIGUSR1 signal to child.

```
if ( ( pid = fork() ) < 0)
   printf("error\n");
else if (pid == 0) {
   /* child does whatever is necessary …*/
   WAIT_PARENT( );
   /* child continues ….*/
   exit(0);
}
/* parent does what ever is necessary …*/
TELL_CHILD(…);
/* parent continues …*/
exit(0);
```

   [Hint: Install a signal handler before fork for SIGUSR1. Declare a flag and set it to 0. Set this flag to 1 in signal handler. In function WAIT_PARENT child may check the value of this flag and may continue only when flag becomes 1.]

2. Login as root user, write and compile the following program.
   int main(){
       printf("uid=%d, euid=%d\n", getuid(), geteuid());
   }
   Set the permissions of the executable file so that anyone can execute it. Set setuid bit ON. Execute the program as an unprivileged user A.
   Now set setuid bit OFF. Execute the program as the unprivileged user A. Is the output different?

3. Create a file "myfile" as the user A (see Q2 above) and set its permissions to 0700.

4. Modify the program in Q2 above to do the following:
   (a) Open file "myfile" and print the file descriptor returned by function open.
   (b) Call setuid function and give user id of user A (in Q2 above) as input.
   (c) Call seteuid function and give user id of the user A (in Q2 above) as input.
   (d) Print user id and effective user id after calling setuid and seteuid functions.
   (e) Open file "myfile" and print the file descriptor returned by function open.

   Execute the program as root user and study the output.