NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Farah Naz 23-NTU-CS-1152

SECTION SE: 5th(A)

LAB MANUAL

SUBMITTED TO:

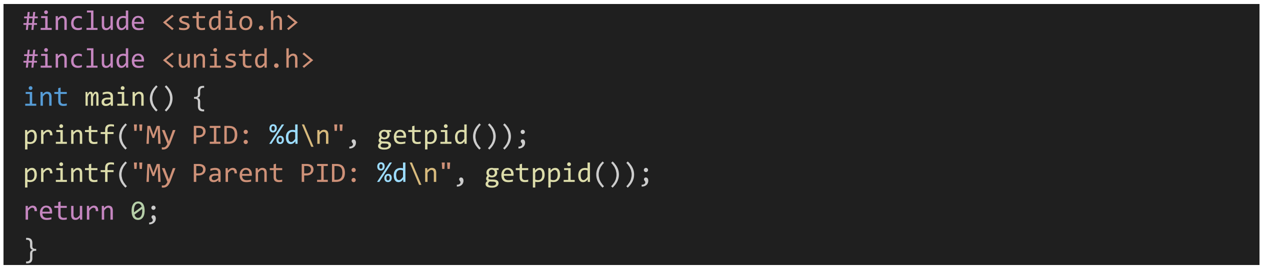
Sir Nasir Mehmood

SUBMISSION DATE:

10-3-2025

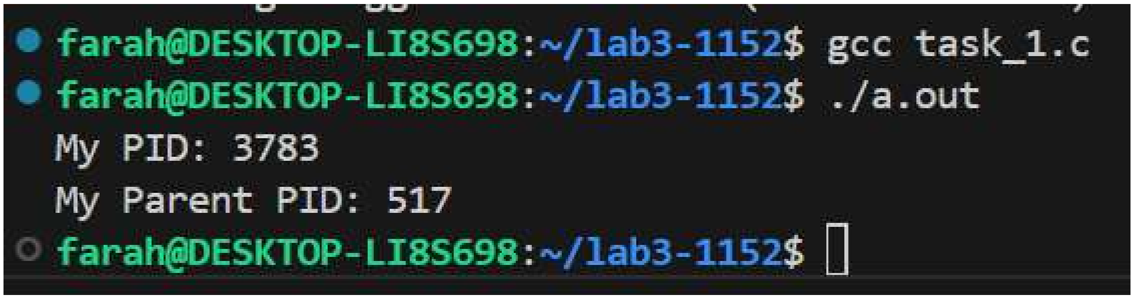
Task 1:

Code:



```
 #include <stdio.h>
 #include <unistd.h>
 int main() {
 printf("My PID: %d\n", getpid());
 printf("My Parent PID: %d\n", getppid());
 return 0;
 }
```

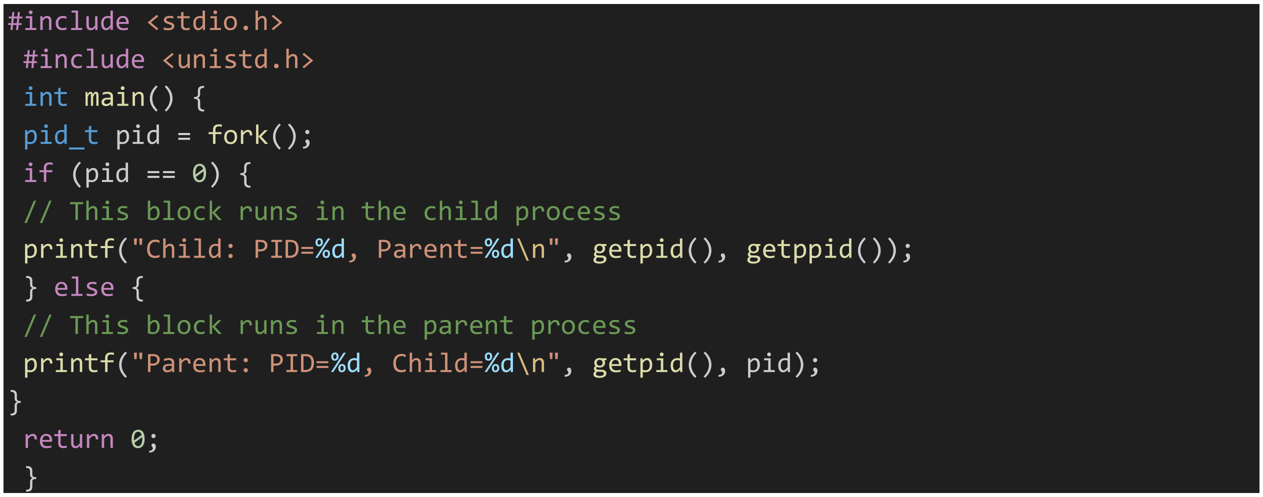
Output:



```
● farah@DESKTOP-LI8S698:~/lab3-1152$ gcc task_1.c
● farah@DESKTOP-LI8S698:~/lab3-1152$ ./a.out
  My PID: 3783
  My Parent PID: 517
○ farah@DESKTOP-LI8S698:~/lab3-1152$ 
```

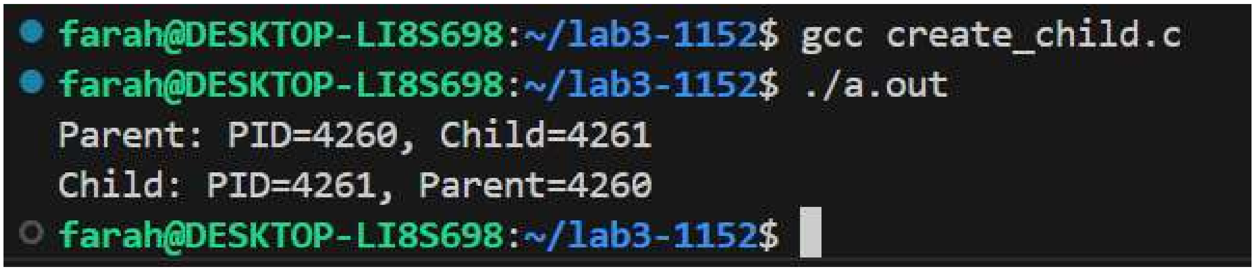
Task 2:

Code:



```
#include <stdio.h>
 #include <unistd.h>
 int main() {
 pid_t pid = fork();
 if (pid == 0) {
 // This block runs in the child process
 printf("Child: PID=%d, Parent=%d\n", getpid(), getppid());
 } else {
 // This block runs in the parent process
 printf("Parent: PID=%d, Child=%d\n", getpid(), pid);
}
 return 0;
 }
```

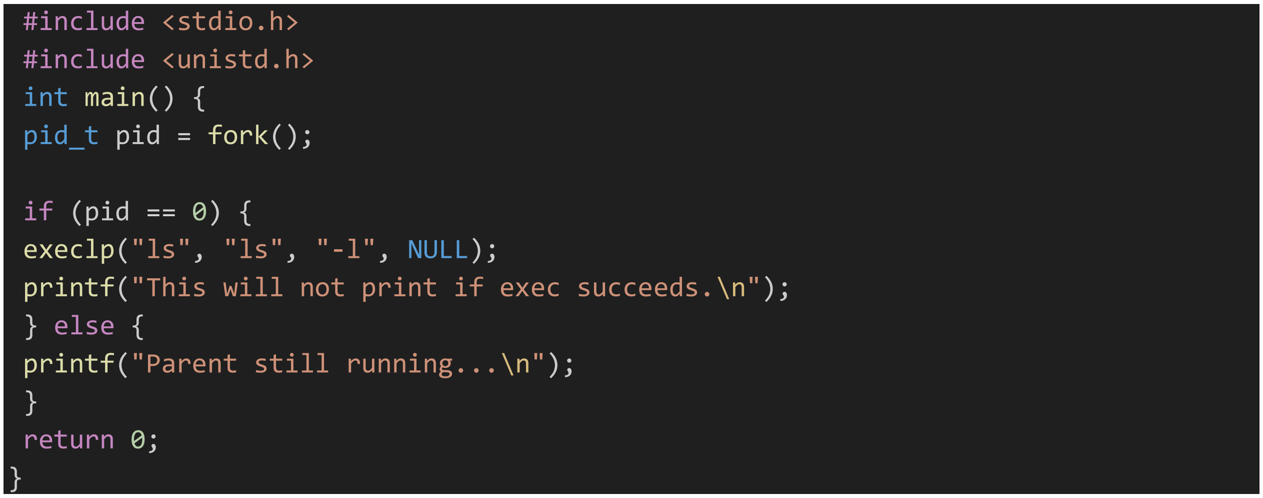
Output:



```
● farah@DESKTOP-LI8S698:~/lab3-1152$ gcc create_child.c
● farah@DESKTOP-LI8S698:~/lab3-1152$ ./a.out
  Parent: PID=4260, Child=4261
  Child: PID=4261, Parent=4260
○ farah@DESKTOP-LI8S698:~/lab3-1152$ 
```

Task 3:

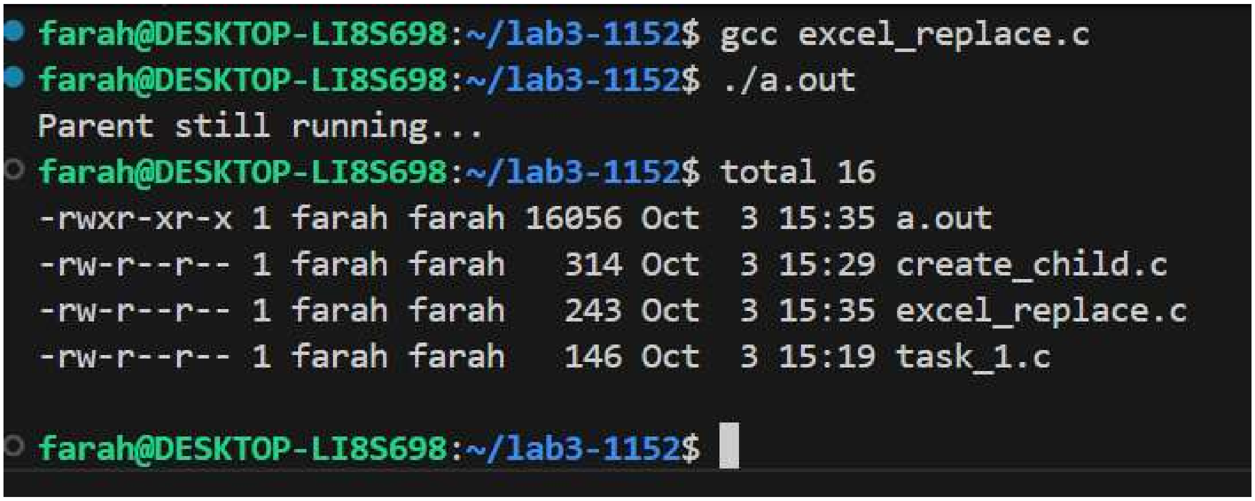
Code:



```
#include <stdio.h>
#include <unistd.h>
int main() {
pid_t pid = fork();

if (pid == 0) {
execlp("ls", "ls", "-l", NULL);
printf("This will not print if exec succeeds.\n");
} else {
printf("Parent still running...\n");
}
return 0;
}
```

Output:

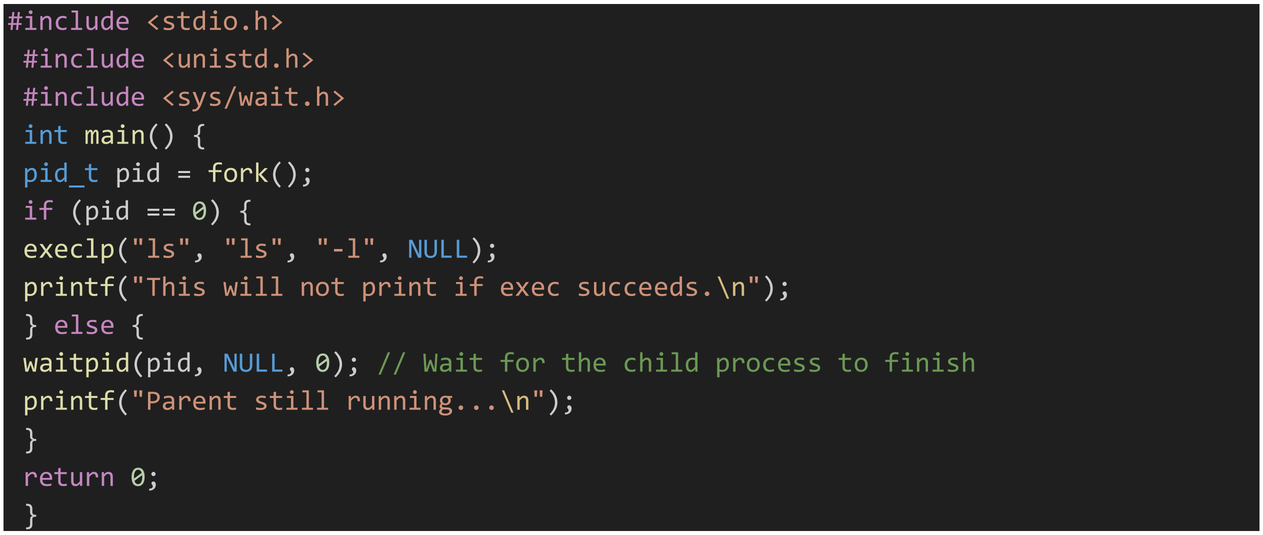


```
● farah@DESKTOP-LI8S698:~/lab3-1152$ gcc excel_replace.c
● farah@DESKTOP-LI8S698:~/lab3-1152$ ./a.out
  Parent still running...
○ farah@DESKTOP-LI8S698:~/lab3-1152$ total 16
  -rwxr-xr-x 1 farah farah 16056 Oct  3 15:35 a.out
  -rw-r--r-- 1 farah farah   314 Oct  3 15:29 create_child.c
  -rw-r--r-- 1 farah farah   243 Oct  3 15:35 excel_replace.c
  -rw-r--r-- 1 farah farah   146 Oct  3 15:19 task_1.c

○ farah@DESKTOP-LI8S698:~/lab3-1152$ 
```

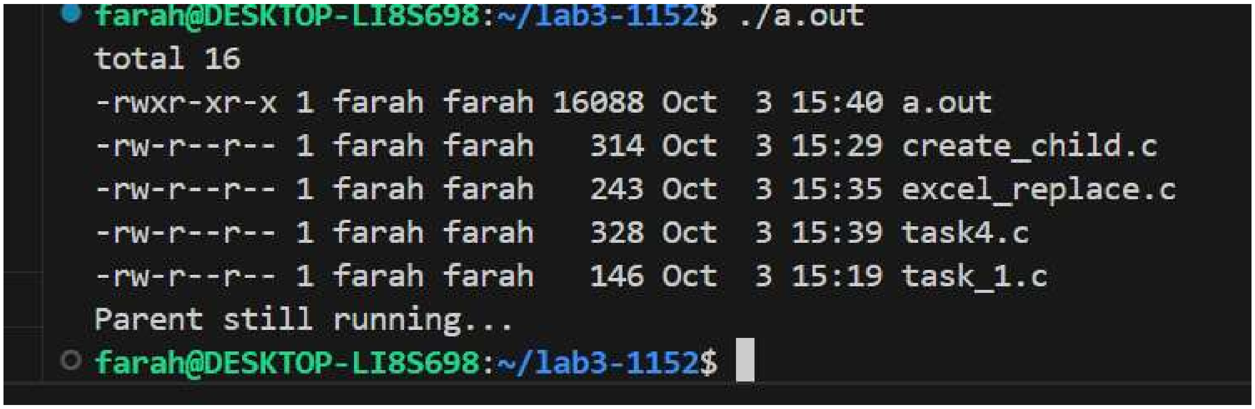
Task 4:

Code:



```
#include <stdio.h>
 #include <unistd.h>
 #include <sys/wait.h>
 int main() {
 pid_t pid = fork();
 if (pid == 0) {
 execlp("ls", "ls", "-l", NULL);
 printf("This will not print if exec succeeds.\n");
 } else {
 waitpid(pid, NULL, 0); // Wait for the child process to finish
 printf("Parent still running...\n");
 }
 return 0;
 }
```

Output:



```
farah@DESKTOP-LI8S698:~/lab3-1152$ ./a.out
total 16
-rwxr-xr-x 1 farah farah 16088 Oct  3 15:40 a.out
-rw-r--r-- 1 farah farah   314 Oct  3 15:29 create_child.c
-rw-r--r-- 1 farah farah   243 Oct  3 15:35 excel_replace.c
-rw-r--r-- 1 farah farah   328 Oct  3 15:39 task4.c
-rw-r--r-- 1 farah farah   146 Oct  3 15:19 task_1.c
Parent still running...
farah@DESKTOP-LI8S698:~/lab3-1152$
```