Operating Systems Lab
Koushik Sahu
118CS0597
Date: 7th September, 2020

Q1.

Output:

This line will be printed twice if parent and child process has the same program counter after fork is called
I am child process
Global variable:21 Local variable value:20
Address of Global variable:0x5644cf4ee010 Address of local_variable:0x7ffc93f08390
This line will be printed twice if parent and child process has the same program counter after fork is called
I am parent process
Global variable:21 Local variable value:20
Address of Global variable:0x5644cf4ee010 Address of local_variable:0x7ffc93f08390

```
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ ./a.out
This line will be printed twice if parent and child process has the same program counter after fork is called
I am child process
Global variable:21      Local variable value:20
Address of Global variable:0x555f7821c010       Address of local_variable:0x7fffddfe2ed0
This line will be printed twice if parent and child process has the same program counter after fork is called
I am parent process
Global variable:21      Local variable value:20
Address of Global variable:0x555f7821c010       Address of local_variable:0x7fffddfe2ed0
```

Q2.

Input: 7
Output:

Enter the number of processes to be created in chain
7
Parent process:16171 Child process:16172
Parent process:16172 Child process:16173
Parent process:16173 Child process:16174
Parent process:16174 Child process:16175
Parent process:16175 Child process:16176
Parent process:16176 Child process:16177
Parent process:16177 Child process:16178

```
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ gcc 07092020118CS0597Q2.c
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ ./a.out
Enter the number of processes to be created in chain
7
Parent process:13556    Child process:13878
Parent process:13878    Child process:13879
Parent process:13879    Child process:13880
Parent process:13880    Child process:13881
Parent process:13881    Child process:13882
Parent process:13882    Child process:13883
Parent process:13883    Child process:13884
```

Q3. Input: 2 3
3 3
1 10
Output:

Note: If the depth is n, the longest path from root to leaf would have n processes
Eg: If depth is 1, then there is 1 process no matter what degree is passed

---

Enter the required depth: 2
Enter the required degree: 3
Parent process:16486 Child process:16487
Parent process:16486 Child process:16488
Parent process:16486 Child process:16489
Parent process:15989 Child process:16486

Enter the required depth: 3
Enter the required degree: 3
Parent process:16499 Child process:16500
Parent process:16499 Child process:16501
Parent process:16499 Child process:16502
Parent process:16491 Child process:16499
Parent process:16503 Child process:16504
Parent process:16503 Child process:16505
Parent process:16503 Child process:16506
Parent process:16491 Child process:16503
Parent process:16507 Child process:16508
Parent process:16507 Child process:16509
Parent process:16507 Child process:16510
Parent process:16491 Child process:16507
Parent process:15989 Child process:16491

Enter the required depth: 1
Enter the required degree: 10
Parent process:15989 Child process:16517

```
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ gcc 07092020118CS0597Q3.c
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ ./a.out
--------------------------------------------------------------------------------
Note: If the depth is n, the longest path from root to leaf would have n processes
Eg: If depth is 1, then there is 1 process no matter what degree is passed
--------------------------------------------------------------------------------

Enter the required depth: 2
Enter the required degree: 3
Parent process:14544    Child process:14554
Parent process:14544    Child process:14555
Parent process:14544    Child process:14556
Parent process:12717    Child process:14544
```

```
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ ./a.out
-------------------------------------------------------------------------------------------------
Note: If the depth is n, the longest path from root to leaf would have n processes
Eg: If depth is 1, then there is 1 process no matter what degree is passed
-------------------------------------------------------------------------------------------------

Enter the required depth: 3
Enter the required degree: 3
Parent process:14580     Child process:14581
Parent process:14580     Child process:14582
Parent process:14580     Child process:14583
Parent process:14578     Child process:14580
Parent process:14584     Child process:14585
Parent process:14584     Child process:14586
Parent process:14584     Child process:14587
Parent process:14578     Child process:14584
Parent process:14588     Child process:14589
Parent process:14588     Child process:14590
Parent process:14588     Child process:14591
Parent process:14578     Child process:14588
Parent process:12717     Child process:14578
```

```
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$ ./a.out
-------------------------------------------------------------------------------
Note: If the depth is n, the longest path from root to leaf would have n processes
Eg: If depth is 1, then there is 1 process no matter what degree is passed
-------------------------------------------------------------------------------

Enter the required depth: 1
Enter the required degree: 10
Parent process:12717    Child process:14596
koushik@koushik:~/Documents/code/nitr/os_lab/07092020$
```