

## MultiProcessing Assignments 2 :-

### Answers :-

#### 1)Output:-

```
nikki@INLEN6239004542: ~/Systemprogramming/Assignment2/question1
nikki@INLEN6239004542:~/Systemprogramming/Assignment2/question1$ ./bin/sysPrg1 input.txt

Child Created!
Child's Process Id: 354
Parent's Process Id: 353

Writing to file [Child Process]...
(Enter "-1" to exit)
Hi I am Nikki
Hello World
Thank you
-1

Child exited with status: 0
Reading from file [Parent Process]...
Hi I am Nikki
Hello World
Thank you
```

- A file created named "input.txt".

```
nikki@INLEN6239004542:~/Systemprogramming/Assignment2/question1$ ls -R
.:
bin  inc  input.txt  obj  script  src

./bin:
sysPrg1

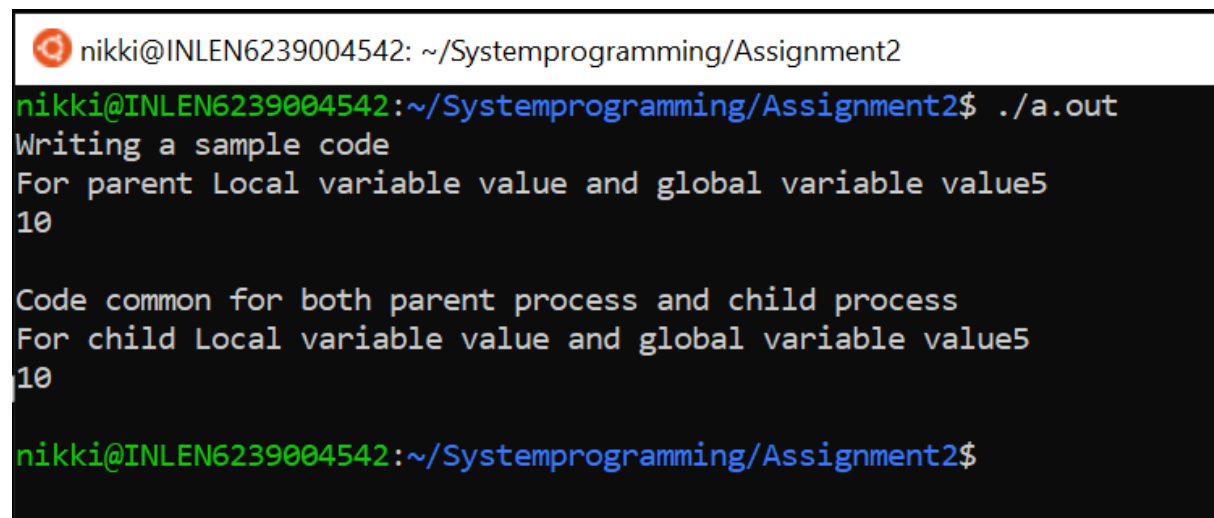
./inc:
myHeader.h

./obj:
sysPrg1.o

./script:
Makefile

./src:
sysPrg1.cpp
nikki@INLEN6239004542:~/Systemprogramming/Assignment2/question1$
```

2) (a) The output is 5 and 10 for both parent and child processes.

A terminal window with a dark background and light-colored text. The prompt is 'nikki@INLEN6239004542: ~/Systemprogramming/Assignment2'. The user enters './a.out'. The output shows 'Writing a sample code', 'For parent Local variable value and global variable value5', and '10'. Then it shows 'Code common for both parent process and child process', 'For child Local variable value and global variable value5', and '10'. The prompt returns at the end.

```
nikki@INLEN6239004542: ~/Systemprogramming/Assignment2
nikki@INLEN6239004542:~/Systemprogramming/Assignment2$ ./a.out
Writing a sample code
For parent Local variable value and global variable value5
10

Code common for both parent process and child process
For child Local variable value and global variable value5
10

nikki@INLEN6239004542:~/Systemprogramming/Assignment2$
```

(b) Yes, The both values are same for parent and child processes.

(c) Yes , order of execution is same everytime and there is no impact in output when we excute everytime , it is same everytime.

(d) Process executes last statement and then asks the operating system to delete it using the exit()system call. So, here we used exit() in child process, therefore the cout statement of child process not displayed only parent process cout is displayed.