**Program 1**

**What is the output of the following code:**

void forkexample(){

int x = 1;

pid\_t p = fork();

if(p<0){

perror("fork fail");

exit(1); }

else if (p == 0)

printf("Child has x = %d\n", ++x);

else

printf("Parent has x = %d\n", --x);}

int main(){

forkexample();

return 0; }

Answer: Parent has x = 0

Child has x = 2

**Program 3**

**What is the output of the following code?**

#include <stdio.h>

#include <unistd.h>

int main() {

if (fork() || fork())

fork();

printf("osw\n");

return 0; }

**Program 4 (a) How many times iter will be printed and how many processes are create by following code including program process.**

int main(){

int pid,pid2;

pid=fork();

if(pid) {

pid2=fork();

printf("iter");

}

else {

printf("iter");

pid2=fork();

}

return 0;}

**(b) Modify the above code by adding a newline chracter "\n" in the else block printf statement. Then find how many times iter will be displayed.**

**Program 5: Check the output and state the reason for this output.**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main(void){

printf("Fork create many confusion");

fork();

printf("We suppose to read and Practice\n");

return 0;

}

**Modify the above code by adding a newline chracter "\n" in the printf() before fork().**

**Program 6**

**Check how many ITER will be printed for the following code snippet?**

#include <stdio.h>

#include <unistd.h>

int main(void) {

fork();

fork();

fork();

fork();

printf("ITER\n");

printf("ITER\n");

printf("ITER\n");

sleep(4);

return 0;

}

Program\_File: fk2.c