(I)

Test drive the various examples of fork that have been discussed so far in the class. Understand the output generated and infact try editing those codes with possible combinations such as changing the parent / child definitions, point of forking, etc. this will help you to better appreciate the working of fork system call

Also practice the below given set of examples with the same objective as above.

As far as this set goes you can generate the report (only) for the below problems, containing the output generated, your manual trace (as we did in the class), your reasonings / understanding of the output.

II Fork Practice Set

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
int main()
{
pid_t pid;
pid=fork();
if (pid!=0)
fork();
fork();
printf("Count \n");
return 0;
}
```

```
int main()

{

printf(“OS \n");

fork();

fork();

fork();

}
```

```
int main ()
{
printf("This will be printed ?.\n");
fork();
printf("This will be printed ?.\n");
fork();
printf("This will be printed ? .\n");
fork();
printf("This will be printed ?\n");
return 0;
}
```

```
int main ()
{
printf(“A \n”);
fork();
printf(“B\n”);
return 0;
}
```

```
int main()
{
printf(“OS ");
fork();
fork();
fork();
}
```

```
int main ()
{
printf(“A”);
fork();
printf(“B”);
return 0;
}
```

Express the following in a process tree setup and also write the C code for the same setup

1 forks 2 and 3

2 forks 4 5 and 6

3 forks 7

4 forks 8

5 forks 9