OS LAB TASK #06

Name: Khushal Das
Roll No: 22P-9341
Section : BSE-5B

Exercise #3.1.1

```
fast@HALAB-12:~/Desktop/main$ vim myfirst.c
fast@HALAB-12:~/Desktop/main$ gcc  myfirst.c -o myfirst
fast@HALAB-12:~/Desktop/main$ ./myfirst

This is C Programming in Ubuntu


Here in this program we will run linux commands


PID of myfirst.c: 8624
PPID of myfirst.c: 6380
fast@HALAB-12:~/Desktop/main$
fast@HALAB-12:~/Desktop/main$
```

PIP is: 8624
PPID is: 6380

3.2.1 Creation States:

```
fast@HALAB-12:~/Desktop/main$ vim process.c
fast@HALAB-12:~/Desktop/main$ gcc process.c -o process
fast@HALAB-12:~/Desktop/main$ ./process
Process PID   9146       PPID   6380
Process PID   9147       PPID   9146
fast@HALAB-12:~/Desktop/main$
```

Q1 How many processes are created?
Ans: 2 processes are created.

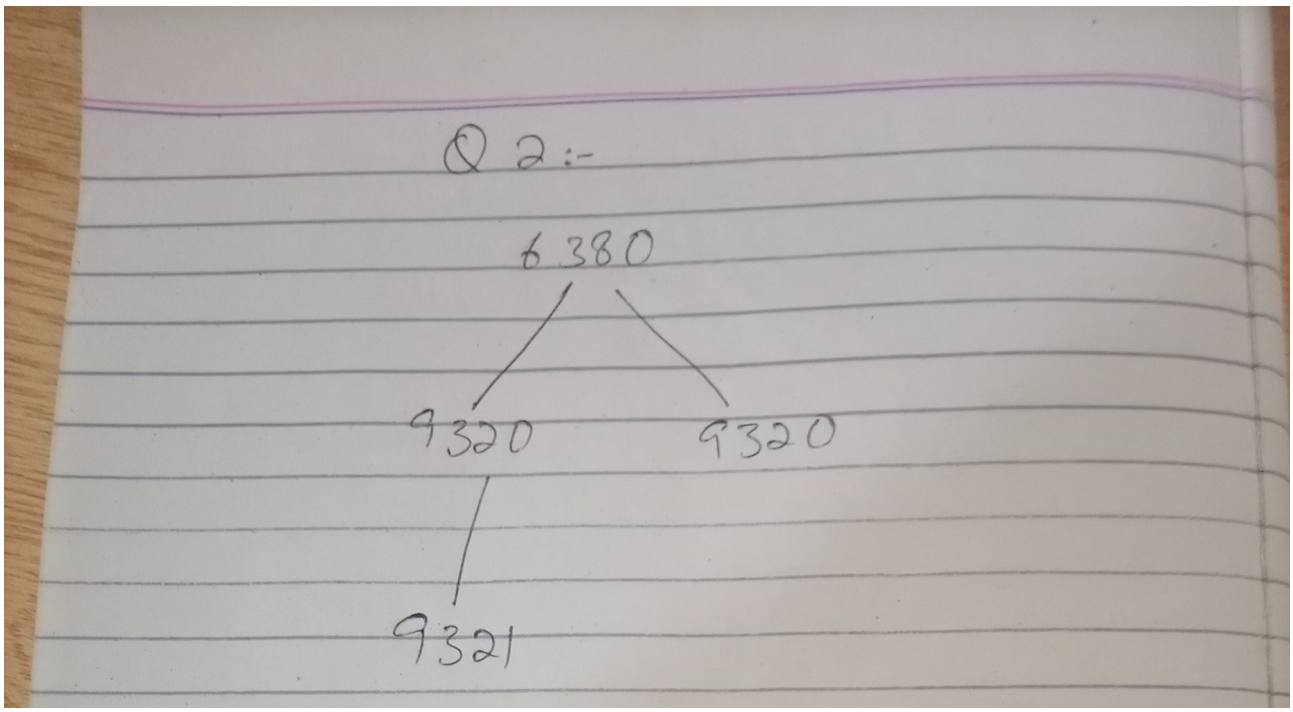
OS LAB TASK #06

Q2:

```
fast@HALAB-12:~/Desktop/main$ vim process1.c
fast@HALAB-12:~/Desktop/main$ gcc process1.c -o process1
fast@HALAB-12:~/Desktop/main$ ./process1
Process PID   9320        PPID   6380
Process PID   9321        PPID   9320
Process PID   9322        PPID   9320
Process PID   9323        PPID   9321
fast@HALAB-12:~/Desktop/main$
```

How many processes does it show this time?

Ans: 04



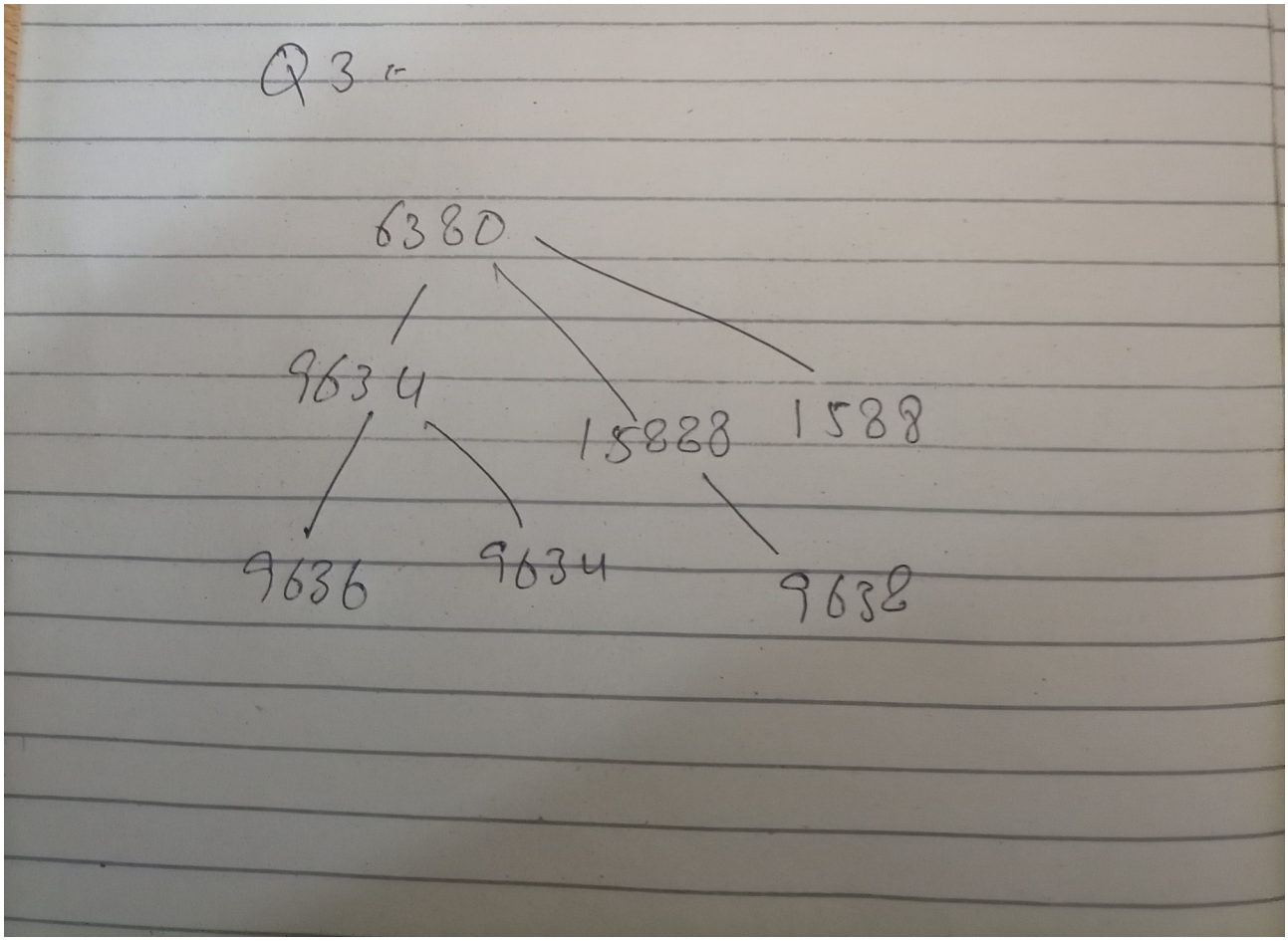
OS LAB TASK #06

Q3:

```
fast@HALAB-12:~/Desktop/main$ vim process2.c
fast@HALAB-12:~/Desktop/main$ gcc process2.c -o process2
fast@HALAB-12:~/Desktop/main$ ./process2
Process PID   9634        PPID   6380
Process PID   9635        PPID   9634
Process PID   9636        PPID   9634
Process PID   9637        PPID   9634
Process PID   9639        PPID   9636
Process PID   9638        PPID   1588
fast@HALAB-12:~/Desktop/main$ Process PID   9640        PPID   1588
Process PID   9641        PPID   9638
```

How many processes does it show?

Ans: 08



OS LAB TASK #06

Q#04:

What is going to happen?
Ans: Infinity loop will be executed and 2 power 100 fork created so all the memory will be allocated when processes are created.

Does your OS Crash? **YES**
Does your Program Crash? **YES**
Can you modify your code to count the total number of fork()s made? **Yes**

OS LAB TASK #06

Q5 Can a Ho be output before a He? Why?

Ans:

OS LAB TASK #06

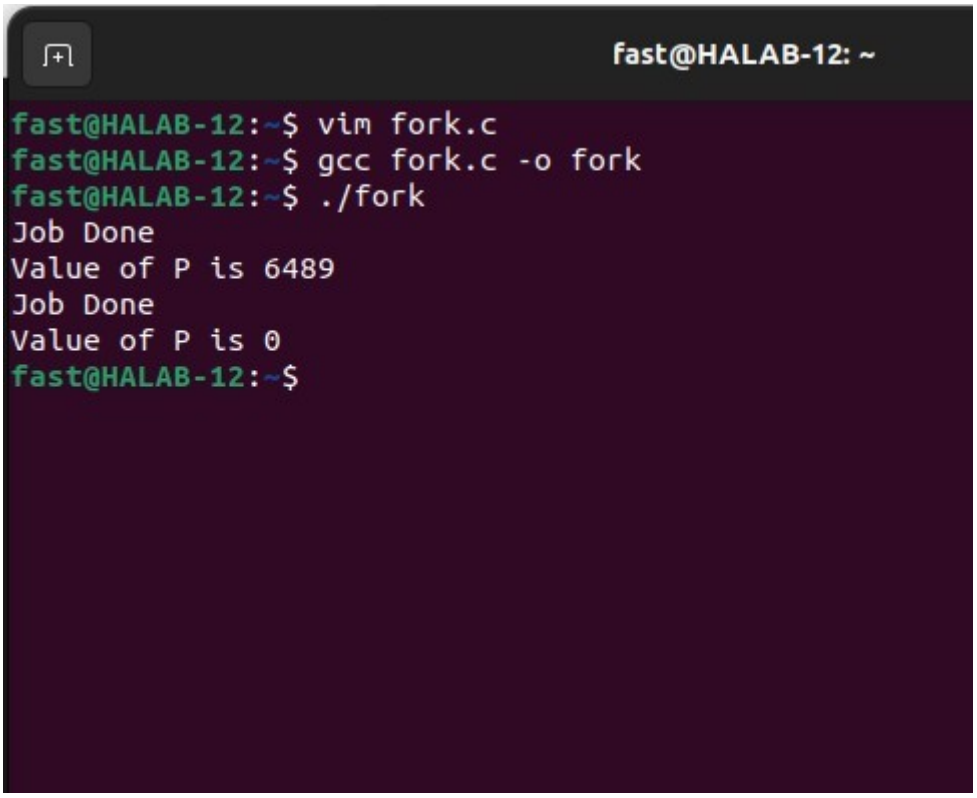
Q1: `p = fork()` return the PID value for process.

Q2: `printf` is part of the C Standard Library, included in `stdio.h`.

Q3: Output appears twice because both the parent and child processes execute the `printf` statement after the fork.

Q4:

Output:



Q4: What would happen if we don't use the If/Else conditions and immediately write the printf() statements?

Ans:

Using `if`/`else` statements helps us clearly use of the parent and child processes, it shows output to read and understand. Without them, both processes will execute the same code.

3.2.1.1 Exercise 1

```
khushaldas@khushaldas-Latitude-E7240:~/Desktop/main$ ./fork
Original Process, pid = 15187
Parent PID = 15187, Child ID = 15188
Child PID = 15188, PPID = 15187
```

```
khushaldas@khushaldas-Latitude-E7240: ~
khushaldas@khushaldas-Latitude-E7240:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
```

```
khushaldas@khushaldas-Latitude-E7240:~$ ps
    PID TTY          TIME CMD
  15263 pts/5    00:00:00 bash
  15643 pts/5    00:00:00 ps
khushaldas@khushaldas-Latitude-E7240:~$
```

3.2.1.2 Exercise 2:

```
khushaldas@khushaldas-Latitude-E7240:~/Desktop/main$ gcc fork3.c -o fork3
khushaldas@khushaldas-Latitude-E7240:~/Desktop/main$ ./fork3
Parent PID: 16345
C1 PID: 16346, Parent PID: 16345
C2 PID: 16347, Parent PID: 16345
C1.1 PID: 16348, Parent PID: 16346
C2.1 PID: 16349, Parent PID: 16347
C1.2 PID: 16350, Parent PID: 16346
C2.2 PID: 16351, Parent PID: 16347
```