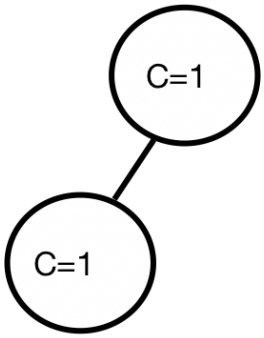
BLG312E - Computer Operating Systems
Assisgnment1

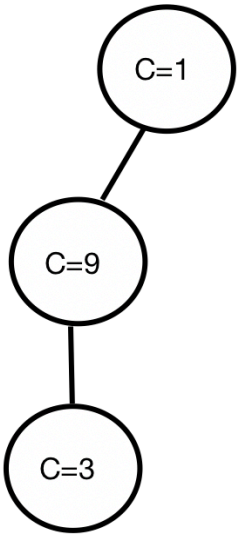
Ekin Çelebi
150160152

Question 1

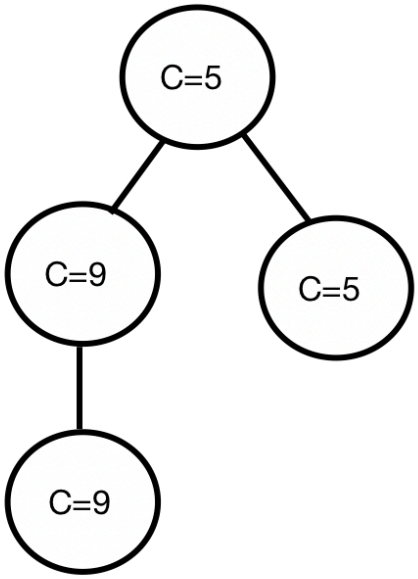
```
int main(int argc, char* argv[])
{
        int c = 0;
        int child = fork();
        c++;
```




```
if(child == 0)
{
    child = fork();
    c += 2;
    if(child) c = c*3;
}
```



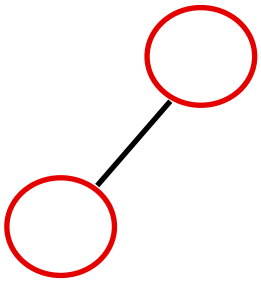
```
        else
        {
                c += 4;
                fork();
        }

        return 0;
}
```



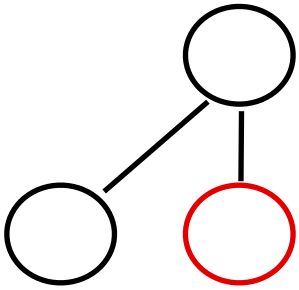
1.a) There will be 4 processes created.
1.b) There are 4 ‘c’ variables generated at the end
1.c) The latest values stored in the processes are shown above.

Details for Question 2

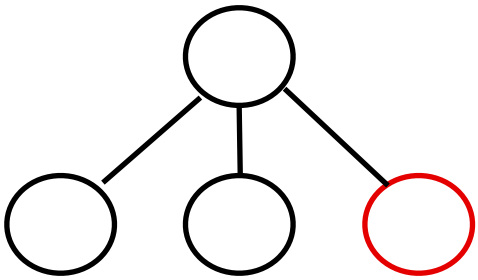
Call fork() this creates a parent and a child process.



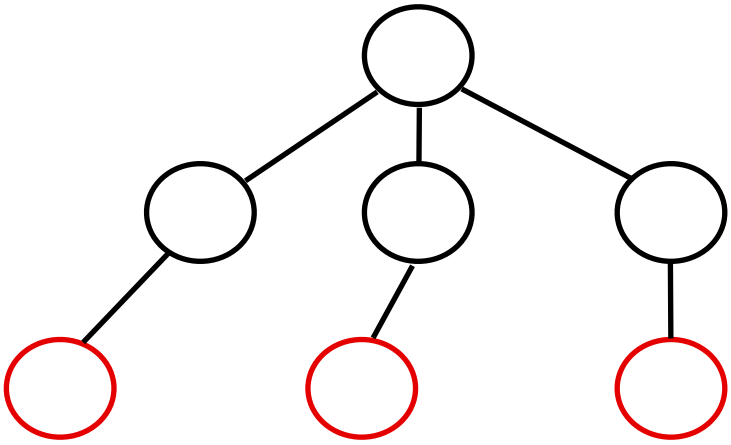
Call fork() from the parent(level 1) node.



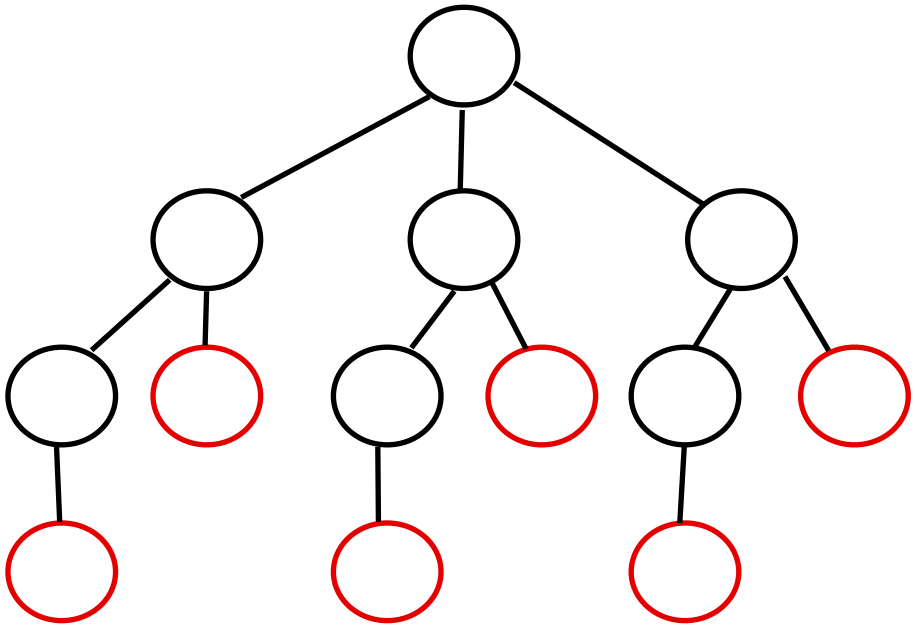
Again check from the parent(level 1) node and call fork() again.



This time check for child processes(level 2) and call fork() for each child.



Call fork() to create processes for both child(level 3) and parent(level 2) nodes.



Compilation

[celebie16@ssh ~]$ gcc oshw1.c
[celebie16@ssh ~]$ ./a.out

Program output

```
PID: 31372, Children PID: 31373, 31374, 31375, level: 1
PID: 31373, Children PID: 31376, 31378, level: 2
PID: 31378, Children PID: 0, level: 3
PID: 31376, Children PID: 31379, level: 3
PID: 31379, Children PID: 0, level: 4
PID: 31375, Children PID: 31380, 31383, level: 2
PID: 31377, Children PID: 31382, level: 3
PID: 31374, Children PID: 31377, 31381, level: 2
PID: 31381, Children PID: 0, level: 3
PID: 31380, Children PID: 31384, level: 3
PID: 31383, Children PID: 0, level: 3
PID: 31382, Children PID: 0, level: 4
PID: 31384, Children PID: 0, level: 4
```

Tree of outputs