1) Consider the following program

```
1.      int c1 = fork();
2.      if(c1 == 0 && fork())      ——→ (A)
                       c2
3.      {
4.          printf("Catch me\n");
5.      }
6.      else if(c1 > 0)    ——→ (P) ——→ awaits for (A)
7.      {
8.          waitpid(c1, NULL, 0);
9.          printf("If you can\n");
10.     }else              ——→ (B)
11.     {
12.         printf("Now you see me\n");
13.     }
```

a) Draw the process tree for the above program [5]
b) What will be the output of the program? [3]

False: &&

True   &&   [fork()]

$c_1 > 0$

(P)

fork()

$c_2 > 0$ (A) $c_1 = 0$

(B) $c_1 = 0$ [copy from parent A]
     $c_2 = 0$
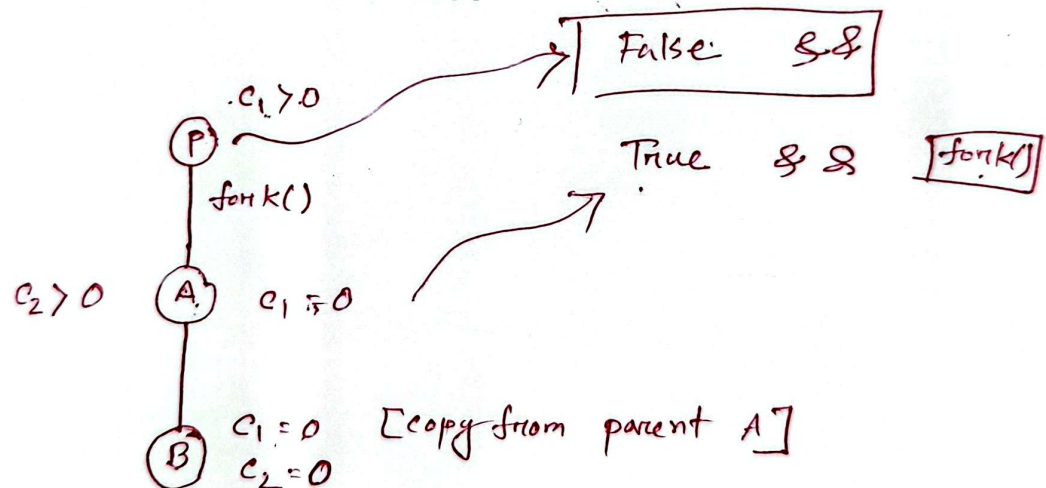
```
Catch me
If you can
```
Now you see me

Now you see me
Catch me
If you can

Catch me
Now you...
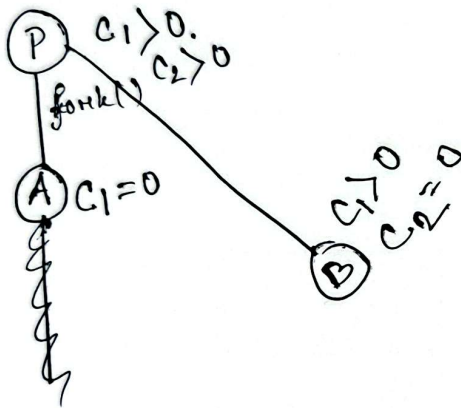If you can

3) Consider the following program:

```
1.      if(fork() && fork())
2.      {
3.          printf("I must print\n");
4.      }
5.      else
6.      {
7.          printf("Will I be printed again?\n");
8.      }
```

a) Draw the process tree for the above program [5]
b) What will be the output of the program? [3]

Process tree: P with $c_1 > 0$, $c_2 > 0$; fork() to A with $c_1 = 0$; to B with $c_1 > 0$, $c_2 = 0$.

Output:

I must print.

will I be printed again?

will I be printed again?