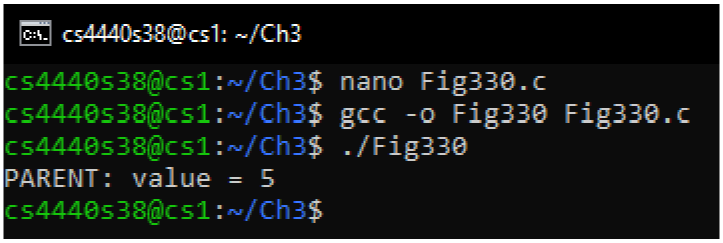
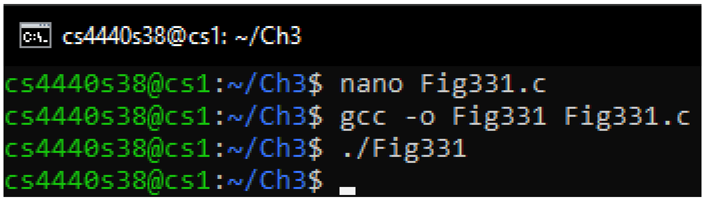
3.1 Using the program shown in Figure 3.30, explain what the output will be at LINE A.



```
cs4440s38@cs1:~/Ch3$ nano Fig330.c
cs4440s38@cs1:~/Ch3$ gcc -o Fig330 Fig330.c
cs4440s38@cs1:~/Ch3$ ./Fig330
PARENT: value = 5
cs4440s38@cs1:~/Ch3$
```

The output will be five because the child updates the value copy.

3.2 Including the initial parent process, how many processes are created by the program shown in Figure 3.31?

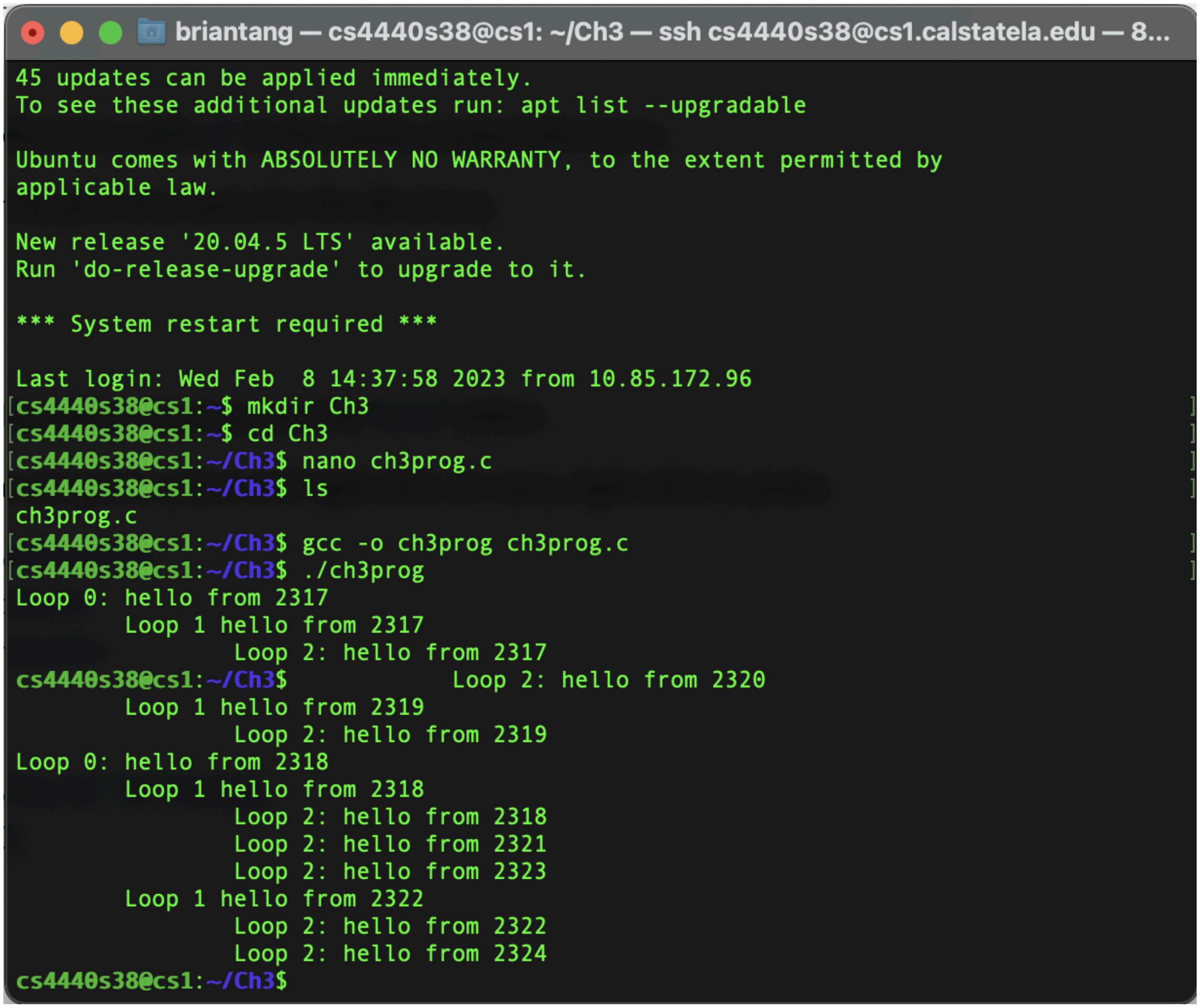


```
cs4440s38@cs1:~/Ch3$ nano Fig331.c
cs4440s38@cs1:~/Ch3$ gcc -o Fig331 Fig331.c
cs4440s38@cs1:~/Ch3$ ./Fig331
cs4440s38@cs1:~/Ch3$ _
```

Since each fork function is 2^n, and we have three forks. To put this together, it will look like 2^3.
That equals to a total of 8 processes, 2^3 = 8.

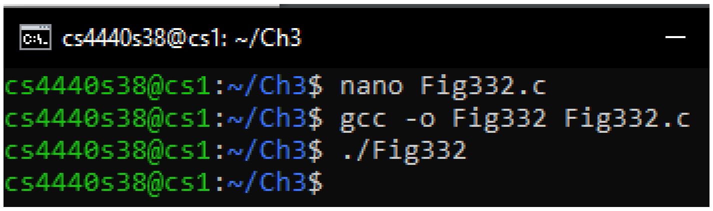
3.11 Including the initial parent process, how many processes are created by the program shown: Fig 3.32 (Fig 3.21)

```
#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main()

{
        int i;

        for (i=0; i<3; i++)

        {

                fork();

                if (i == 0) printf("Loop 0: hello from %d\n", getpid());

                if (i == 1) printf("\tLoop 1 hello from %d\n", getpid());

                if (i == 2) printf("\t\tLoop 2: hello from %d\n", getpid());

        }

        return 0;

}
```



briantang — cs4440s38@cs1: ~/Ch3 — ssh cs4440s38@cs1.calstatela.edu — 8...

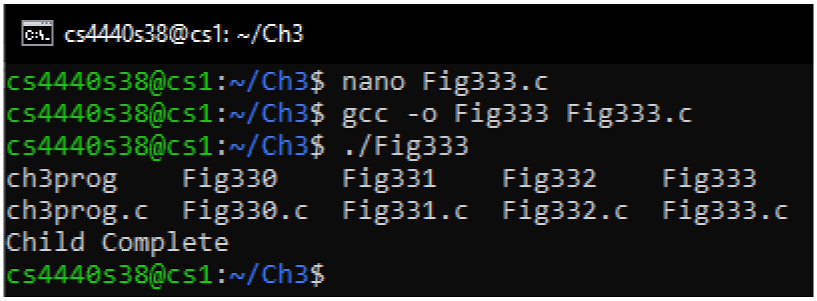
```
45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***

Last login: Wed Feb  8 14:37:58 2023 from 10.85.172.96
cs4440s38@cs1:~$ mkdir Ch3
cs4440s38@cs1:~$ cd Ch3
cs4440s38@cs1:~/Ch3$ nano ch3prog.c
cs4440s38@cs1:~/Ch3$ ls
ch3prog.c
cs4440s38@cs1:~/Ch3$ gcc -o ch3prog ch3prog.c
cs4440s38@cs1:~/Ch3$ ./ch3prog
Loop 0: hello from 2317
        Loop 1 hello from 2317
                Loop 2: hello from 2317
cs4440s38@cs1:~/Ch3$            Loop 2: hello from 2320
        Loop 1 hello from 2319
                Loop 2: hello from 2319
Loop 0: hello from 2318
        Loop 1 hello from 2318
                Loop 2: hello from 2318
                Loop 2: hello from 2321
                Loop 2: hello from 2323
        Loop 1 hello from 2322
                Loop 2: hello from 2322
                Loop 2: hello from 2324
cs4440s38@cs1:~/Ch3$
```



```
cs4440s38@cs1:~/Ch3$ nano Fig332.c
cs4440s38@cs1:~/Ch3$ gcc -o Fig332 Fig332.c
cs4440s38@cs1:~/Ch3$ ./Fig332
cs4440s38@cs1:~/Ch3$
```

This is a fork function. Again, it will be 2^n. Since we have to go through the loop four times, n would be four.
So the conclusion would be 2^4 = 16 processes.

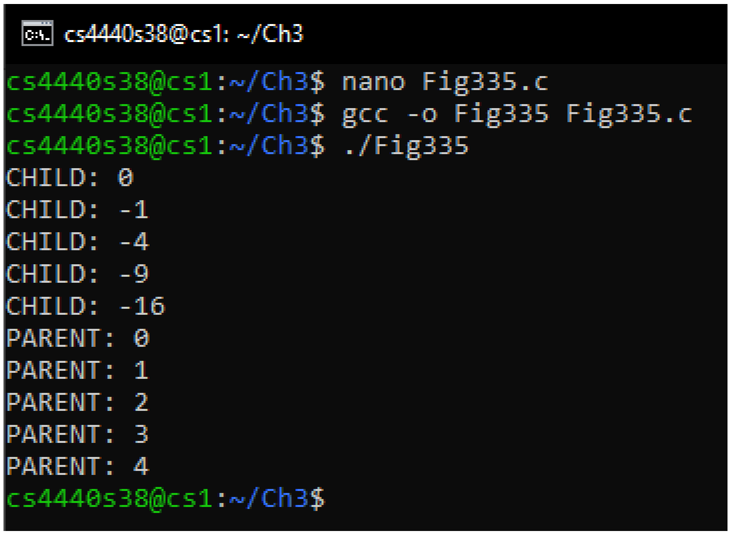
3.12 Explain the circumstances under which the line of code marked printf("LINE J") in Figure 3.33 (Fig 3.22) will be reached.



```
cs4440s38@cs1:~/Ch3$ nano Fig333.c
cs4440s38@cs1:~/Ch3$ gcc -o Fig333 Fig333.c
cs4440s38@cs1:~/Ch3$ ./Fig333
ch3prog    Fig330    Fig331    Fig332    Fig333
ch3prog.c  Fig330.c  Fig331.c  Fig332.c  Fig333.c
Child Complete
cs4440s38@cs1:~/Ch3$
```

This code reads and find all the file names in your directory correspondingly. If the code doesnt find the file names, it wouldn't read it, so "LINE J" will be executed.

3.16 Using the program shown in Figure 3.35 (Fig 3.24), explain what the output will be at lines X and Y.



```
cs4440s38@cs1:~/Ch3$ nano Fig335.c
cs4440s38@cs1:~/Ch3$ gcc -o Fig335 Fig335.c
cs4440s38@cs1:~/Ch3$ ./Fig335
CHILD: 0
CHILD: -1
CHILD: -4
CHILD: -9
CHILD: -16
PARENT: 0
PARENT: 1
PARENT: 2
PARENT: 3
PARENT: 4
cs4440s38@cs1:~/Ch3$
```

The output of line X prints out each child of 0, -1, 4, -9, and -16.
The output of line Y prints out each parent of 0, 1, 3, and 4.