

```

#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[]) {

    int num;
    pid_t pid;

    for (num= 0; num< 3; num++) {
        pid= fork();
        printf ("Soy el proceso de PID %d y mi padre tiene %d de PID.\n",
                getpid(), getppid());
        if (pid!= 0)
            break;
        srandom(getpid());
        sleep (random() %3);
    }
    if (pid!= 0)
        printf ("Fin del proceso de PID %d.\n", wait (NULL));

    return 0;
}

```



```

#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

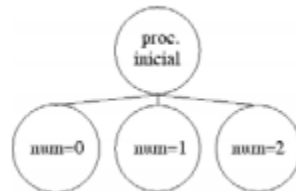
int main (int argc, char *argv[]) {

    int num;
    pid_t pid;

    srandom(getpid());
    for (num= 0; num< 3; num++) {
        pid= fork();
        printf ("Soy el proceso de PID %d y mi padre tiene %d de PID.\n",
                getpid(), getppid());
        if (pid== 0)
            break;
    }
    if (pid== 0)
        sleep(random() %5);
    else
        for (num= 0; num< 3; num++)
            printf ("Fin del proceso de PID %d.\n", wait (NULL));

    return 0;
}

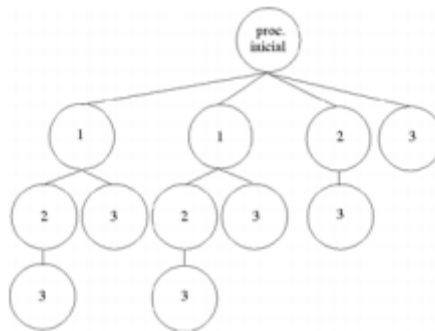
```



```

for (num= 0; num< 2; num++) {
    nuevo= fork(); /* 1 */
    if (nuevo== 0)
        break;
}
nuevo= fork(); /* 2 */
nuevo= fork(); /* 3 */
printf("Soy el proceso %d y mi padre es %d\n", getpid(), getppid());

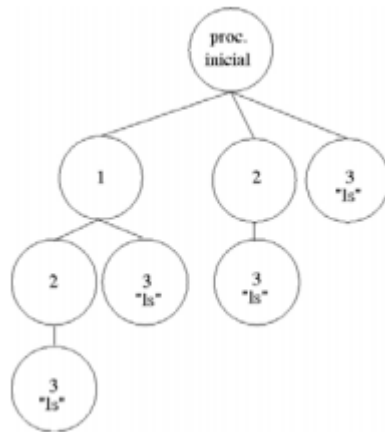
```



```

for (num= 1; num<= n; num++){
    nuevo= fork();
    if ((num== n) && (nuevo== 0))
        execlp ("ls", "ls", "-l", NULL);
}

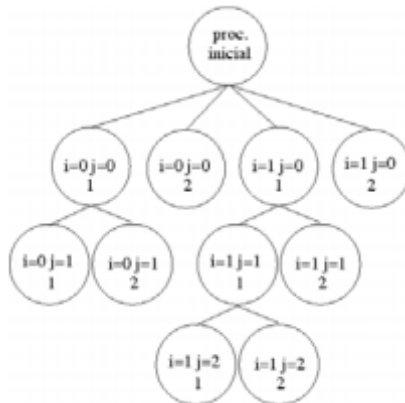
```



```

for (i= 0; i< 2; i++) {
    pid= getpid();
    for (j= 0; j< i+2; j++) {
        nuevo= fork();    /* 1 */
        if (nuevo!= 0) {
            nuevo= fork(); /* 2 */
            break;
        }
    }
    if (pid!= getpid())
        break;
}

```



```

#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

#define L1 2
#define L2 3

int main (int argc, char *argv[]) {

    int cont1, cont2;
    pid_t pid;

    for (cont2= 0; cont2< L2; cont2++) {
        for (cont1= 0; cont1< L1; cont1++) {
            pid= fork();
            if (pid== 0)
                break;
        }
        if (pid!= 0)
            break;
    }
    return 0;
}

```

