

1. Suponga que ejecutamos el siguiente código en C:

```
1 int main() {
2     if (fork() != fork()) {
3         pid_t t = fork();
4         printf("Artorias\n");
5         if (t > 0) {
6             printf("Ornstein\n");
7         }
8         else if (t == 0) {
9             printf("Smough\n");
10            exit(0);
11        }
12        else {
13            printf("Hydra\n");
14            exit(0);
15        }
16    }
17    printf("Gwyn\n");
18 }
```

- (a) **(20 puntos)** ¿Cuántos procesos se crean en total? Justifique su respuesta con un dibujo que represente el árbol de procesos generado.
- (b) **(20 puntos)** Indique cuántas veces se imprime cada una de las posibles salidas del código.

2. Dado el siguiente código en C:

```
1 int main() { // codigo1
2     pid_t r[3];
3     for (int i = 0 ; i < 3 ; i++) {
4         r[i] = fork();
5         if (r[i] == 0) {
6             printf("Ramen\n");
7             execlp("./codigo2", "", NULL);
8         }
9     }
10    sleep(10);
11    printf("Sushi\n");
12    for (int i = 0 ; i < 3 ; i++) {
13        kill(r[i], SIGKILL);
14    }
15 }
```

```
1 int main() { // codigo2
2     sleep(666);
3     fork();
4     printf("Brisket\n");
5     exit(0);
6 }
```

- (a) **(20 puntos)** Suponga que existe una justa asignación de tiempo de CPU entre los procesos del sistema. ¿Cuántos procesos se crean en total? ¿Cuáles son las posibles salidas? Justifique su respuesta.

