1. Suponga que ejecutamos el siguiente código en C:

```
int main() {
      if (fork() != fork()){
           pid_{-}t t = fork();
           printf("Artorias\n");
           if (t > 0)
                printf("Ornstein\n");
           else if (t==0)
               printf("Smough\n");
               exit (0);
10
11
                printf("Hydra\n");
               exit(0);
14
15
16
      printf("Gwyn\n");
17
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:

```
int main(){ //codigo1
      pid_t r[3];
      for (int i = 0; i < 3; i++) {
          r[i] = fork();
          if (r[i] == 0)
                                              int main() { //codigo2
              printf("Ramen\n");
                                                    sleep (666);
              execlp("./codigo2","",NULL);
                                                    fork();
                                                    printf("Brisket\n");
                                                    exit(0);
      sleep (10);
10
      printf("Sushi\n");
11
      for (int i = 0; i < 3; i++){
          kill (r[i], SIGKILL);
13
14
15
```

(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.