Problema 1 (3 puntos).
Hemos escrito el siguiente programa en UNIX:

primer.c:
```
#include <stdio.h>

main ( argc, argv )
int argc;
char *argv[];
{
  int fd[2];

     pipe( fd );
     switch( fork() ){
       case -1:
         fprintf(stderr, "Error en fork.");
         exit(1);
       case 0:
         close( fd[0] );
         close( 1 );
         dup( fd[1] );
         close( fd[1] );
         execlp( "cat", "cat", argv[1], 0 );
       default:
         close( fd[1] );
         close( 0 );
         dup( fd[0] );
         close( fd[0] );
         wait(NULL);
       }
     execlp( "sort", "sort", 0 );
}
```

Explica detalladamente como se comportará el ejecutable "`primer`" cuando lo utilicemos desde nuestra cuenta en "`merlin`" pasándole como parámetro cualquiera de estos 2 ficheros:

```
merlin_$  ls -l
total 20
-rw-r-----  1 teresa    arqcomp     8360  Jul 30      12:28 segun.c
-rw-r-----  1 teresa    arqcomp     440   Jul 30      12:23 primer.c
```

¿Tiene el mismo efecto la ejecución de "`primer segun.c`" que la de "`primer primer.c`"?. Si no es así, explica por qué.

Problema 2 (3 puntos).

Diseñar un programa llamado "`replicar`" en lenguaje C, usando sólo llamadas al sistema UNIX y cuyo funcionamiento describimos a partir de los siguientes ejemplos de utilización:

merlin_$ `replicar F1`

Copiará lo que le introducimos por teclado sobre el fichero `F1` (cuyo nombre le ha sido pasado como primer parámetro) y además lo mostrará por pantalla.

merlin_$ `replicar F3 < F1 > F2`

Creará dos ficheros, `F2` y `F3`, ambos con el mismo contenido que `F1`.

merlin_$ `replicar`

No hará nada. Termina devolviendo un código de error de valor 2, ya que le falta el parámetro.

Problema 3 (4 puntos).

Hemos escrito el siguiente programa en UNIX:

programa.c:

```
#include <signal.h>
#include <stdio.h>

void funcion(n)
int n;
{
      signal(n+1, funcion);
      write(2, "UN MENSAJE\n", 11);
}

main()
{
int pid;
      signal(SIGUSR1, funcion);
      if((pid = fork()) == 0){
            pid = getppid();
            kill( pid, SIGUSR1 );
      }
      else {
            kill (getpid(), SIGUSR2);
      }
}
```

Después de obtener el ejecutable "`programa`" y de forma aleatoria, podemos obtener como resultados de la ejecución de "`programa`" lo siguiente:

```
merlin_$ programa        merlin_$ programa merlin_$ programa
UN MENSAJE               UN MENSAJE              User signal 2
UN MENSAJE               UN MENSAJE              merlin_$
UN MENSAJE               merlin_$
merlin_$
```

Explica por qué se obtienen estos 3 tipos de resultados.