

Tarea 4

Problema 4.

El código:

```
/*CODIGO 1*/
#define testError(err) if (err) { printf("Hay un error fatal\n"); exit(1); }
    if (...)
        testError(err)
    else
        ...
```

es equivalente al código:

```
/*CODIGO 2*/
    if (...)
        if (err)
        {
            printf("Hay un error fatal.\n");
            return 1;
        }
    else
        ...
```

Es decir, al usar el `define` dentro de la estructura condicional tendremos que el comando *else* será parte ahora del *if* definido en el *define*. De esta manera no se podrá tener lo que se supone que se quiere.

Una manera de arreglar el problema es no definir el *define* y escribiendo el código como en el segundo código. Otra manera es no usar el *define* definido en el primer código en una estructura condicional como la del segundo código. La tercera manera es declarando la estructura condicional del *define* con un *else* (código corregido 1) o usando los operadores `? :`. Otra manera es cambiar la condición de *if* por su negativo (código corregido 2):

```
/*CODIGO CORREGIDO 2*/
#define testError(err) if (err) { printf("Hay un error fatal\n"); exit(1); }
else ( blablabla );
```

Tarea 4

```
if ( ... )
    testError(err)
else
    ...
```

Esta es otra solución:

```
/*CODIGO CORREGIDO 1*/
#define testError(err) if (err) { printf("Hay un error fatal\n"); exit(1); }
if ( ! ... )
    ...
else
    testError(err)
```