

Ejercicio a entregar:

1-. Crear un fichero utils.h y otro fichero util.c. En él, debemos crear las siguientes funciones:

1.1: Función cuenta_numero_letras

```
int cuenta_numero_letras(char cadena[], char letra)
```

Esta función recorrerá la cadena de caracteres y comparará si es igual a dicha letra. Si es así aumentará un contador que devolveremos al finalizar de recorrer.

1.2 Función compara_cadena

```
int compara_cadena(char *cadena1, char *cadena2)
```

Esta función devolverá 1 si la dos cadenas son iguales; en caso contrario, devolverá 0.

Nota: Usar la función strcmp. Buscar funcionamiento de la función vía Google.

1.3 Función tipo_de_letra

```
int tipo_de_letra (char letra)
```

Esta función usará el argumento letra comprobar si:

- Si esa letra es una 'a', devolver un 1
- Si esa letra es una 'c', devolver un 2
- Si esa letra es una 'e', devolver un 3
- Si no es ninguna de las anteriores devolver un 0.

Se valorará el uso de switch.

1.4 Función primera_letra

```
int primera_letra(char cadena[], char letra)
```

Esta función recorrerá la cadena de caracteres, devolverá la posición de la primera letra que sea igual al argumento letra.

Se valorará el uso de break y continue.

1.5 Función cambia_letras

```
int cambia_letras(char cadena[], char letraoriginal, char letracambiar)
```

Esta función recorrerá la cadena de caracteres y comprobará si la cadena de caracteres contiene la letra que pasamos por el argumento letraoriginal.

- Si es así, sustituirá la letra de la cadena de caracteres por la letra a cambiar (letracambiar).
- En caso contrario, seguirá buscando.

1.6 Función suma_numeros

```
int suma_numeros (int n, int m)
```

Esta función devolverá la suma entre n y m.

1.7 Función resta_numeros

```
int resta_numeros (int n, int m)
```

Esta función devolverá la resta entre n y m.

1.8 Función multiplica_numeros

```
int multiplica_numeros (int n, int m)
```

Esta función devolverá la multiplicación entre n y m.

Todas estas funciones que hemos declarado deben ser públicas y contenidas en el fichero utils.h

2-. Crear un fichero llamado main.c en el cuál se incluirá la cabecera utils.h y para utilizar las funciones declaradas.

2.1-. Dicho fichero contendrá tres funciones estáticas:

- static int test1(void): Esta función comprobará las funciones 1.1, 1.2 y 1.3 declaradas anteriormente. Llamaremos a dichas funciones y utilizando la sentencia assert comprobar si el valor devuelto es el esperado.
- static int test2(void): Esta función comprobará las funciones 1.4 y 1.6 declaradas anteriormente. Llamaremos a dichas funciones y utilizando la sentencia assert comprobar si el valor devuelto es el esperado.
- static int test3(void): Esta función comprobará las funciones 1.7 y 1.8 declaradas anteriormente. Llamaremos a dichas funciones y utilizando la sentencia assert comprobar si el valor devuelto es el esperado.

2.2-. Debemos añadir la función main:

```
void main(void)
```

Esta función llamará a las tres funciones estáticas declaradas anteriormente. También comprobará el funcionamiento de la función 1.5. Para realizar dicha comprobación, deberá imprimir el mensaje antes y después de ejecutar dicha función.

EXTRAS

3-. Usar automake y autoconf para configurar y compilar el proyecto existente.

4-. Generar un parche con todo el código desarrollado y enviarlo al correo <alvaro@soleta.eu>. Guardar dicho fichero también en la carpeta del proyecto.

Los ejercicios deben ser subidos a la cuenta personal de cada alumno de GitHub. El ejercicio debe estar subido en dicha cuenta como día límite, el miércoles de la semana que viene. Para toda duda o problema con los ejercicios, tienen la opción de ponerse en contacto con el profesor a partir del correo <alvaro@soleta.eu> o por la lista de programación en C. Se valorará positivamente tanto realizar las preguntas por dicha lista como la resolución de ellas por parte de algún/alguna compañera/o.