

Ejercicios de programación en lenguaje C

Estos programas pueden probarlos primero en la PC usando <http://cpp.sh/> o <http://onlinegdb.com/> (elegir lenguaje C y presionar run).

Los usuarios de Linux Pueden usar la plantilla de Embedded IDE "gcc-exec" y probarlo en sus propias PC. Los de Windows hay que instalar más herramientas para compilar C para PC así que usen el compilador on line.

Ejercicios básicos de C

1. Suponga que una variable declarada como

```
uint16_t temperatura_sensor;
```

contiene la salida de un sensor de temperatura con una resolución de 10 bits y un rango de 0°C a 800 °C. Esto es, el valor `temperatura_sensor = 0` corresponderá a la medida de 0 °C mientras que `temperatura_sensor = 1023` corresponderá a la medida de 800 °C.

Sería deseable disponer de una nueva variable: `temperatura_grados` que almacene la medida directamente en grados para poder procesarla y mostrarla por UART directamente.

a. Escriba el programa que transforma el valor entregado por el sensor a grados Celcius y lo almacena en `temperatura_grados`.

b. Pruebe distintos valores de `temperatura_sensor` y conviértalos a `temperatura_grados`.

Imprima los valores por UART. Por ejemplo

```
temperatura_sensor = 0      --> temperatura_grados = 0°C  
temperatura_sensor = 128    --> temperatura_grados = 100°C  
temperatura_sensor = 512    --> temperatura_grados = 400°C  
temperatura_sensor = 1023   --> temperatura_grados = 800°C
```

c. Sabiendo que el rango de la variable `temperatura_sensor` admite valores menores que 0 y mayores que 1023, busque una solución para detectar estos casos e indicarle al usuario de que hay una falla en el sensor.

2. Operadores.

a. Indique el resultado de las siguientes expresiones:

```
(1 > 2) || (1 < 2)
(1 > 2) && 3
1 + 2 / 3 * 4 - 5
```

b. Considerando: `int a=1, b=2, c=3`; diga cuánto valen cada una de las siguientes expresiones:

```
a < b && c < b
a < b && c
a <= c || b > c
a != b - 1 || b > 2 || a + c > 2
```

3. Bucles 1

Realice una tabla utilizando `printf()` en la UART con el siguiente encabezado:

```
x          x al cuadrado      x al cubo
```

- a) complete con valores enteros de x entre 0 y 20 utilizando un bucle `while`.
- b) complete con valores enteros de x entre 0 y 20 utilizando un bucle `for`.
- c) complete con valores reales de x entre 0 y 10, con incrementos de 0.5 utilizando un bucle `while`.
- d) complete con valores reales de x entre 0 y 10, con incrementos de 0.5 utilizando un bucle `for`.

Nota: utilice el caracter '\t' que realiza la función de tab para ordenar correctamente las tablas en la Terminal. Revisar si lo soporta la terminal serie que utiliza.

4. Bucles 2

Realice los siguientes "dibujos" en la Terminal Serie parametrizados según el valor de N y M, según corresponda (usar `#define` para N y M).

- a)
- *
- *
- *
- *
- (N filas)

b)

```
*  
* *  
* * *  
* * * *
```

(N filas)

c)

```
1  
22  
333  
4444
```

(N filas)

d)

```
1  
12  
123  
1234
```

(N filas)

e)

```
* * * * *  
* * * * *  
* * * * *
```

(N filas x M columnas)

f) El cuadrado:

```
* * * * *  
*       *  
*       *  
* * * * *
```

(N filas x M columnas)

g) El triángulo:

```
* * * * *  
*       *  
*   *  
*       *
```

(N filas, con N impar)

Ejercicios de C con Arreglos (vectores o matrices)

Resolver los siguientes utilizando índices de arreglos

- 5.** Realice nuevamente el ejercicio 3 pero esta vez utilice los arreglos `x[]`, `x2[]` y `x3[]` para almacenar primero los valores correspondientes de `x`, `x` al cuadrado y `x` al cubo, respectivamente.
Luego repita la tabla a partir de los arreglos anteriores.
- 6.** Dado un arreglo de 40 caracteres que contiene un string a elección, realizar un programa que cuente la cantidad de caracteres y la cantidad de mayúsculas del string e imprima el resultado usando índice para recorrer el arreglo.
- 7.** Realice un programa que copie el string contenido en el arreglo `Texto1` al arreglo `Texto2` cambiando todas las letras mayúsculas por minúsculas y luego imprima un string debajo del otro en pantalla. Ambos arreglos son de 30 caracteres.
- 8.** Dado cuatro arreglos de 20 caracteres: `Texto1`, `Mayu`, `Minu` y `Noletra`, realizar un programa que a partir del string a elección almacenado en `Texto1`, genere 3 strings: Uno en `Mayu` con todas las mayúsculas, otro en `Minu` todas las minúsculas y otro en `Noletra` con todos los caracteres que no sean letras.
Finalmente deberá imprimir los 4 strings en líneas sucesivas dentro de llaves o corchetes.
- 9.** Dados dos arreglos de 50 caracteres `Tx1` y `Tx2` que contiene strings a elección, realizar un programa que imprima los strings y luego concatene el string contenido en `Tx2` al de `Tx1`. Finalmente Imprimir el string resultante contenido en `Tx1`. Todos los strings deberán imprimirse dentro de llaves o corchetes.
- 10.** Dado un arreglo de 30 caracteres que contiene un string a elección, realizar un programa que cuente e imprimir a la cantidad de vocales de cada tipo. Usar la estructura `switch`.