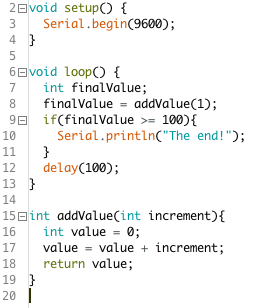
**Arduino parte 2**

**Sebastian Vargas Baena**

**ID 195237**

**P21. Muestre cómo funciona el stack en este programa.**



En el stack se anota la dirección de retorno, en el caso del codigo que estamos estudiando la línea 9, la variable increment también se almacena en el stack, lo mismo para la variable value.

**P22. Invente un programa que haga uso de los siguientes operadores lógicos: &&, ||, ! .**

**P23. Analice el programa que se muestra en el siguiente slide (utilice este enlace para verificar el funcionamiento)**

**P23a. ¿Qué hace la función atoi?**

Convierte una cadena a entero, analiza la cadena e interpreta su contenido como un número entero, que se devuelve como un valor tipo INT, la función descarta primero los espacios en blanco

**P23b. ¿Qué hace el operador %?**

El operador % saca el módulo de una división, calcula el resto de un entero cuando se divide por otro

**P23c. ¿En qué parte de la memoria está myData?**

**P23d. ¿Qué significa char str[ ]?**

Declarar un array de caracteres sin inicialización

**P24. ¿Cómo es posible que una función pueda llamarse con diferentes tipos de parámetros?**

random(max) random(min, max)

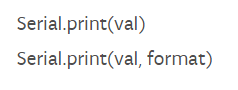
**Parámetros**

* min - límite inferior del valor aleatorio, ambos inclusive (opcional)
* max - límite superior del valor aleatorio, exclusivo

**P25. ¿Para qué sirve la función randomSeed()?**

randomSeed (semilla) inicializa el generador de números pseudo-aleatorio, haciendo que se inicie en un punto arbitrario en una secuencia aleatoria. Si es importante para una secuencia de valores generados por azar () para variar, en posteriores ejecuciones de un boceto, utilice randomSeed () para inicializar el generador de números aleatorios con una entrada bastante aleatorio

**P26. ¿Para que sirve el parámetro format que se muestra en la figura?**

****

**Parámetro**

**Format:** especifica la base numérica (para tipos de datos integrales) o el número de decimales (para tipos de punto flotante)

**P27. ¿Cuál es la diferencia entre las funciones print y write?**

**Print:**

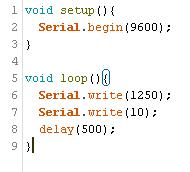
Imprime datos al puerto serie como texto ASCII legible. Este comando puede tomar muchas formas. Los números se imprimen usando un caracter ASCII para cada dígito. Los flotadores se imprimen de manera similar como dígitos ASCII, el impago de dos cifras decimales. Bytes se envían como un solo personaje. Los personajes y las cadenas se envían de inmediato

**Write:**

Escribe los datos binarios al puerto serie. Estos datos se envían como un byte o una serie de bytes; para enviar a los personajes que representan los dígitos de un número utilizan la impresión de la función () en su lugar.

**P28. Explique el funcionamiento del programa que se muestra la figura de la derecha. Utilice como recurso el siguiente URL para convertir el número 1250 a hexadecimal:**

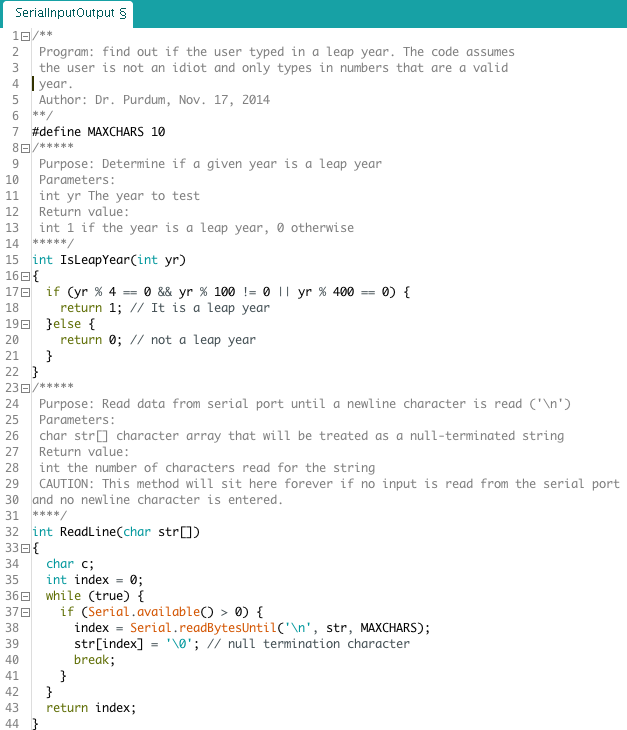
<http://www.binaryhexconverter.com/decimal-to-hex-converter>



Este programa le entra un número entero y lo convierte a un carácter (chart), ejemplo con el numero **1250** el programa escribe la letra **â** o el número **115** y entrega la letra **S.**

**P29. Para discutir en grupo: ¿Qué quiere decir una transmisión asincrónica?**

**Codigo para P 30,31,32**





**P30. Explique para qué sirve la línea 53.**

* If (Serial.available () > 0) { // lee el byte entrante

**Serial.available ():** Obtener el número de bytes (caracteres) disponibles para su lectura desde el puerto serie. Se trata de datos que ya ha llegado y se almacena en la serie de búfer de recepción (que tiene 64 bytes). available () hereda de la Corriente clase de utilidad.

**P31. En la línea 57 explique cuál es el contenido de la variable myData.**

* bufferCount = ReadLine(mydata);

El **readline** función es similar a la **lectura** función, pero permite que toda una cadena a la entrada en lugar de un solo campo. Normalmente, **leer** se detendrá cuando se encuentra con un delimitador.

**P32. En la línea 34 explique cómo funciona la función ReadLine.**

* Chart c;

Lee el valor del Chart c y determina cuando se encuentra en el límite de la función

**P33. ¿Cómo se explica este comportamiento?**

**#define MAXCHARS 10**

**int IsLeapYear(int yr)**

**{**

**if (yr % 4 == 0 && yr % 100 !=0 || yr % 400 == 0){**

**return 1;**

**}else{**

**return 0;**

**}**

**}**

**int ReadLine(char str[])**

**{**

**char c;**

**int index = 0;**

**while(true){**

**if(Serial.available() > 0) {**

**index = Serial.readBytesUntil('\n', str, MAXCHARS);**

**str[index] = '\0';**

**break;**

**}**

**}**

**return index;**

**}**

**void setup()**

**{**

**Serial.begin(9600);**

**}**

**void loop()**

**{**

**if (Serial.available() > 0) {**

**int bufferCount;**

**int year;**

**char myData[MAXCHARS + 1];**

**bufferCount = ReadLine(myData);**

**year = atoi(myData);**

**Serial.print("year: ");**

**Serial.print(year);**

**Serial.print(" is ");**

**if (IsLeapYear(year) == 0){**

**Serial.print("not ");**

**}**

**Serial.println("a leap year");**

**}**

**}**

**P34. ¿Cómo se explica este ERROR?**