INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

Unidad III: Manejo de Funciones Cadenas.

Nombre: Arturo Matamoros Balderas
2019640534
Grupo: 1MV1
Materia: Introducción a la Programación

Introducción

La mayoría de los programas de computo que resuelven problemas reales son mucho mas grandes que los programas que se puedan realizar con operadores de un carácter simple. A través de la practica observamos que la mejor manera de desarrollar y mantener un programa grande es construirlo a partir de piezas pequeñas o módulos, los cuales son mas manejables que el programa original. En esta técnica se denomina divide y vencerás. En esta unidad veremos las características del lenguaje C que faciliten el diseño, la implementación, la operación y el mantenimiento de programas grandes.

Objetivos

- Presentar al lector las funciones matemáticas disponibles en la biblioteca de C.
- Crear y diseñar nuevas funciones.
- Comprender como escribir y utilizar funciones que se invocan a sí mismas
- Comprender el mecanismo utilizado para pasar información entre funciones.
- Comprender como construir programas de manera modular mediante pequeñas piezas llamadas funciones

Desarrollo

Funciones

Una función es un conjunto de instrucciones que se le puede llamar desde el programa principal o desde otras funciones. Las funciones sirven para desarrollar algo en especial, cuando el programador lo necesite.

Para usar funciones, se deben tener en cuenta dos cosas que necesita:

1. La declaración de la función: Esto sirve para que, al compilar el programa, el compilador reconozca que esa función existe; ya que, si se llama desde alguna parte del programa sin haberla declarado o habiéndola declarado, se la declaro mal eso dará error.

Para declararla se sigue la siguiente sintaxis:

¿Qué es prototipo y que son parámetros?

Prototipo de la función: sirve para indicar que va a retornar la función, si va a retornar un entero, un doublé, un char, o simplemente no retoma nada (void). Esto es obligatorio.

Parámetros: son los datos que recibe o que se le envían a la función para que con ellos posiblemente desarrolle algo que se necesite. Esto es opcional.

Entonces, una función para ingresar podría ser así:

Void ingresar ();

Donde se aprecia que no va a retornar nada (por ello tiene void); entonces la función hace lo que tiene que hacer y termina, no devuelve nada.

Un ejemplo de devolución seria en una suma de enteros, ya que se le podría enviar comodato los dos números y la función haría internamente la suma devolviendo la suma de dichos números (si son todos enteros, devolvería un int. Si son enteros y flotantes devolvería unfloat), así:

Int suma enteros (int, int); // suma únicamente enteros, devuelve un entero

Float suma_numeros (float, Float); // suma enteros o flotantes, devuelve un flotante

La definición de la función: Sirve ya para desarrollar la función; es decir ya programar dentro de ella para que haga lo que se necesita. Como consecuencia de que ya está creada se puede usarla, pero si no se hubiese declarado y se desarrolla (hacemos la definición), al compilar dará error y dirá que esa función no fue declarada, que no existe. La definición es muy similar a la declaración, solo que esta vez lleva un ambiente (donde se va a programar,

es decir, las llaves “{…}”)

void ingresar ()

{

//El cuerpo de la función

}

También otra diferencia es que la declaración

lleva “;” (punto y coma) al final, la

definición no lo lleva.

Ejemplo

: Hacer un programa que pida dos datos llamando a una función sin retorno y luego confirme si se ingresaron correctamente.

Solución:

El programa debe usar una función sin retorno, es decir una void. Se deben pedir en esa función dos valores; en este caso se van a pedir dos cosas: nombre y número de la suerte; luego se mostrará el mensaje de que se ingresaron correctamente dichos datos

Cadenas

Una cadena en C++ es un conjunto de caracteres, o valores de tipo char, terminados con el carácter nulo, es decir el valor numérico 0. Internamente, en el ordenador, se almacenan en posiciones consecutivas de memoria. Este tipo de estructuras recibe un tratamiento muy especial, ya que es de gran utilidad y su uso es continuo.

La manera de definir una cadena es la siguiente:

Char<identificador> [<longitud máxima>];

Cuando se declara una cadena hay que tener en cuenta que tendremos que reservar una posición para almacenar el carácter nulo terminador, de modo que, si queremos almacenar la cadena "HOLA", tendremos que declarar la cadena como:

Char Saludos [5];

Las cuatro primeras posiciones se usan para almacenar los caracteres "HOLA" y la posición extra, para el carácter nulo.

También nos será posible hacer referencia a cada uno de los caracteres individuales que componen la cadena, simplemente indicando la posición. Por ejemplo, el tercer carácter de nuestra cadena de ejemplo será la 'L', podemos hacer referencia a él como Saludo [2].

Es muy importante tener presente que, en C++, los índices tomarán valores empezando siempre en cero, así el primer carácter de nuestra cadena sería Saludo [0], que es la letra 'H'.

En un programa C++, una cadena puede almacenar informaciones en forma de texto, como nombres de personas, mensajes de error, números de teléfono, etc.

La asignación directa sólo está permitida cuando se hace junto con la declaración.

El siguiente ejemplo producirá un error en el compilador, ya que una cadena definida de este modo se considera una constante, como veremos en el capítulo de "arrays" o arreglos.

Char Saludo [5];

Saludo=” Hola”

La manera correcta de asignar una cadena es:

char Saludo [5];

Saludo [0] = 'H';

Saludo [1] = 'O';

Saludo [2] = 'L';

Saludo [3] = 'A';

Saludo [4] = 0;

Una segunda opción para declarar una cadena es:

Char Saludo [5]=” HOLA”;

Conclusiones

Las funciones nos sirven para que el código tenga una estructura más estética de tal forma que puede tener un mayor entendimiento, también una de sus funciones principales es cuando una parte del programa se repite en varias secciones de la programación principal esto nos dice que es recomendable usar una función para solo colocar el llamado a esta y ahorrar múltiples líneas de código.

Las cadenas nos son útiles para analizar y trabajar cadenas de caracteres las cuales se evalúan como si fuera un arreglo entonces cada carácter se guarda en un espacio del arreglo y de esta forma se consigue una cadena.

Bibliografía

http://c.conclase.net/curso/?cap=008

https://es.scribd.com/doc/97722120/Funciones-en-Dev-C

https://eperdomo89.wordpress.com/2010/10/17/dev-c-clase16-%E2%80%93-funciones-en-general/