Fundamentos de Programación

NRC: ***200274***

Horario: ***Martes y Jueves de 11:00 AM a 12:55PM***

Nombre: ***De Alba Rodríguez Juan Carlos*** ***Código:*** ***220203307***

Tema: ***Funciones en c***

Fecha: ***06/05/2024***

Funciones en c

Las funciones deben tener una definición y deberían tener una declaración, aunque una definición puede actuar como declaración si la declaración aparece antes de que se llame a la función. La definición de función incluye el cuerpo de la función (el código que se ejecuta cuando se llama a la función).

Una declaración de función establece el nombre, el tipo de valor devuelto y los atributos de una función definida en otra parte del programa. Una declaración de función debe preceder a la llamada a la función.

```
tipo_del_resultado NOMBRE(tipo_param1 param1, tipo_param2 param2, ... )
{
   /* Cuerpo de la función */
}
```

El compilador utiliza el prototipo para comparar los tipos de los argumentos en las llamadas subsiguientes a la función con los parámetros de la función y para convertir los tipos de los argumentos a los tipos de los parámetros cuando sea necesario.

Una llamada de función pasa el control de la ejecución de la función de llamada a la función llamada. Los argumentos, si existe, se pasan por valor a la función llamada. La ejecución de una instrucción return en la función llamada devuelve el control y probablemente un valor a la función de llamada.

La función es la unidad modular fundamental en C. Una función normalmente se diseña para realizar una tarea concreta, y su nombre refleja a menudo esa tarea. Una función contiene declaraciones e instrucciones. Esta sección describe cómo declarar, definir y llamar a funciones de C.

¿Qué se puede hacer con las funciones?

Reutilización de Código:

Una de las principales ventajas de las funciones es que te permiten escribir un bloque de código para realizar una tarea específica y luego reutilizarlo en diferentes partes de tu programa.

Esto ayuda a reducir la duplicación de código y facilita el mantenimiento.

Modularización:

Dividir un programa en funciones más pequeñas y manejables facilita la comprensión y el mantenimiento del código.

Cada función puede encargarse de zuna tarea específica, lo que hace que el programa sea más modular y fácil de mantener y depurar.

Abstracción:

Las funciones te permiten abstraer la implementación de una tarea detrás de una interfaz.

Por ejemplo, puedes tener una función que calcule el promedio de una lista de números sin necesidad de conocer los detalles de cómo se realiza el cálculo.

Claridad y Organización:

El uso de funciones bien nombradas y definidas ayuda a mejorar la claridad y la organización del código.

Testing y Depuración:

Las funciones también facilitan la prueba y depuración del código. Puedes probar cada función por separado para asegurarte de que funciona correctamente antes de integrarla en el resto del programa. Además, si hay un error, es más fácil identificar y corregir problemas en una función individual que en todo el programa.

Ejemplos

```
#include <stdio.h>
float sumar(float a, float b) {
    return a + b;
}
float restar(float a, float b) {
    return a - b;
}
float multiplicar(float a, float b) {
    return a * b;
}
float dividir(float a, float b) {
    if (b != 0)
        return a / b;
    else {
        printf("Error: División por cero.\n");
        return 0;
    }
}
int main() {
    float num1, num2;
    char operador;

    printf("Ingrese una operación (por ejemplo, 5 + 3): ");
    scanf("%f %c %f", &num1, &operador, &num2);

    float resultado;
    switch (operador) {
        case '+':
            resultado = sumar(num1, num2);
            break;
        case '-':
            resultado = restar(num1, num2);
            break;
        case '*':
```

```
            resultado = multiplicar(num1, num2);
            break;
        case '/':
            resultado = dividir(num1, num2);
            break;
        default:
            printf("Operador no válido.\n");
            return 1;
    }
    printf("El resultado es: %.2f\n", resultado);
    return 0;
}
```

Este programa es una calculadora usando funciones, como puedes ver no se hacen todos los cálculos por así decirlo no hace todo el main, solo manda a llamar las funciones el main, por lo cual es mas sencillo en cuestiones de ejecutar.

Si tuviera un problema la suma, vas directamente ala suma, si la resta esta fallando, entonces vas ala resta y lo reparas etcétera, etcétera.

En programas pequeños no se nota tanto la diferencia, pero yo en un programa que hice en java, fue lo mejor hacerlo de manera modular ya que muy rápido encontraba los errores y lo podía solucionar.