

Programación modular

Para simplificar la resolución de los problemas, basado en la programación estructurada, es posible la división del problema PRINCIPAL en pequeños problemas o sub-problemas de fácil solución y mantenimiento. Cada sub-problema se transformará en una FUNCION.

La solución de los problemas se efectúan desarrollando un Algoritmo, el cual estará compuesto por un tronco o "Algoritmo principal", acompañado de una serie de sub-algoritmos, que unidos adecuadamente resuelven el problema.

El algoritmo principal genera el programa principal, que en C es una función llamada "main". Los sub-algoritmos serán funciones con nombres propios, definidos por el programador.

En general un programa, va a constar de un "tronco" ó programa principal y una serie de subprogramas vinculados con éste. Cuando se ejecuta el programa que contiene funciones, se transfiere en cada llamada, el control al subprograma, se ejecutan sus instrucciones y retorna el control al programa llamador, es como si estuviese intercalado dentro del programa principal.

Las funciones constituyen una herramienta esencial en la programación modular. Por un lado, permiten desarrollar sólo una vez procedimientos (rutinas o subrutinas) que son de utilización frecuente o que se utilizan en diferentes lugares de un mismo programa; por otro lado, permiten utilizar mecanismos de abstracción que facilitan la construcción de un programa concentrándose en los aspectos más relevantes y despreocupándose de detalles que pueden ser encarados en otra etapa del diseño.

Una función es, básicamente, un subprograma; por lo tanto, tiene una estructura similar a la de cualquier programa:

Entradas → Proceso → Salida

Sólo que interactúa (se comunica) con otra función o directamente con el programa principal; dicha comunicación se establece a través de argumentos (parámetros) que conforman los datos (entradas) y resultados (salidas). Como cualquier programa, utilizan variables de trabajo denominadas variables locales proveyendo mecanismos de protección que hacen a éstas inaccesibles desde cualquier otro ámbito fuera de la misma función.

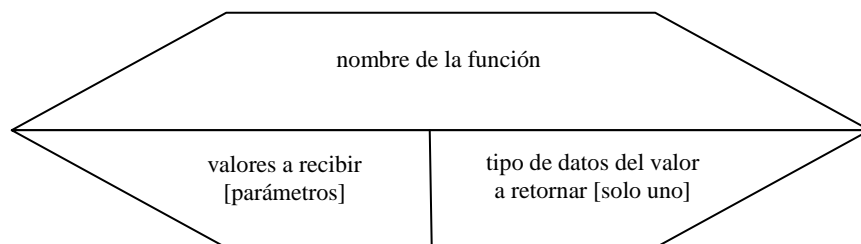
Funciones

Una función es una colección de declaraciones y sentencias que realizan una tarea específica, tiene cuatro componentes.

1. El nombre
2. La información que recibe o toma a través de los parámetros para realizar la tarea. (opcional)
3. La información que devuelve a través de sus parámetros por referencia (opcional)
4. Las sentencias que realizan la tarea
5. El valor que devuelve cuando termina su tarea. (opcional)

Cada función se diagrama por separado, como si fuese otro programa. En la codificación serán incluidas a continuación de la función principal llamada main().

Para la primera línea de la “definición de la función” se utiliza un gráfico especial, que es el siguiente



Para las llamadas a las funciones se utiliza el gráfico que corresponde a una instrucción donde se invocará a la función por su nombre y se le enviarán los datos para utilizarla si es que los requiere.

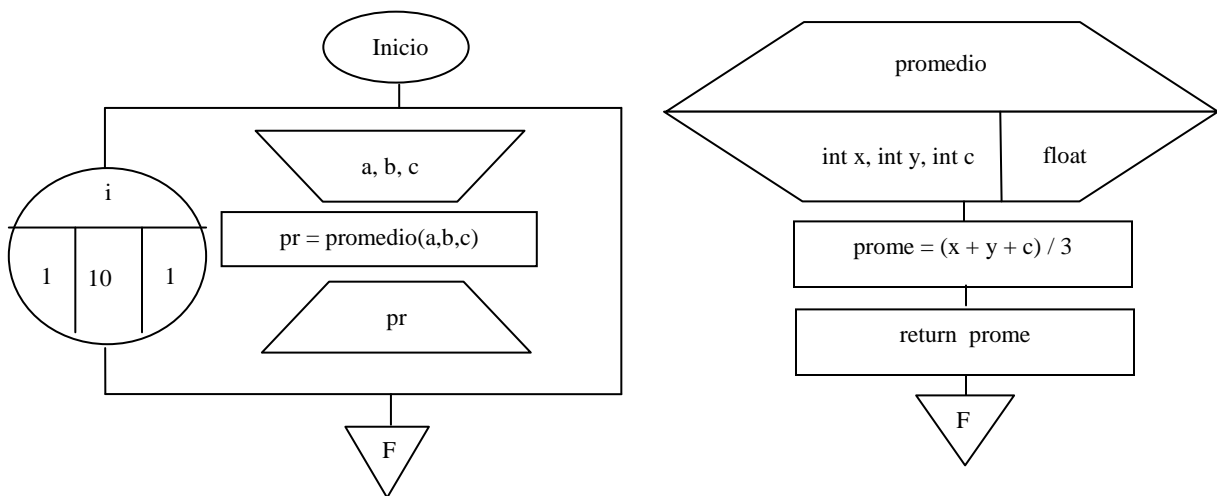
VENTAJAS DEL USO DE FUNCIONES

- Facilita la programación porque divide el problema en subproblemas.
- Simplifica la puesta a punto, ya que puedo corregir o cambiar una función sin tocar el resto del programa. Idem los futuros cambios.
- Permite trabajar simultáneamente a varios programadores.
- Permite la reutilización de estas funciones sin volver a programarlas, solo se utilizan.
- Ahorra memoria, reduce el tamaño ocupado por el programa cuando uso la misma función varias veces.

Parámetros y valor de retorno

A una función opcionalmente se le pueden enviar datos, esos datos son recibidos en la función para ser utilizados dentro de ella. Al especificar la función se detallan esos datos que recibe con su tipo de dato y un identificador que será el nombre de la variable LOCAL a la función donde se guardará el dato recibido. Estos datos que la función recibe se denominan parámetros formales. Veamos el siguiente ejemplo:

Ejemplo: Confeccionar un programa que solicite el ingreso de 10 ternas de valores reales y para cada una de las ternas calcule e informe el promedio de sus valores. Para el cálculo del promedio confeccionar y utilizar una función.



En este ejemplo se ingresan 3 números reales en el programa principal que se almacenan en las variables *a*, *b* y *c*. Luego, se invoca la función *promedio* enviándole como parámetro los tres valores recién leídos. Los parámetros trabajan por posición, es decir, que el primer valor que se indique en la llamada a la función va a ser guardado en una variable llamada *x* dentro de la memoria de la función *promedio*, el segundo la variable *y*, y el tercero en la variable *c*.

Las funciones trabajan con direcciones de memoria totalmente separadas a las del programa principal, al invocar a una función lo que se envía es un valor, es decir, el contenido de una variable y no la variable. Este valor es almacenado en un área distinta de memoria propia de la función. Cada vez que se invoca a una función se reserva un área de memoria separada distinta a la que se utilizó en otras llamadas. Además, al ser un área distinta de memoria es posible utilizar el mismo identificador que en el programa principal pero teniendo en cuenta que aún así es una variable diferente.

Suponga que el usuario ingresa los valores 6, 9 y 10 en el programa principal en las variables *a*, *b* y *c* respectivamente. Al invocar la función lo que se envían son esos números que se COPIAN en los parámetros de la función *promedio*, por cada parámetro se crea automáticamente una variable local a la función por lo que no deben volver a definirse.

La función promedio tiene 4 variables ya que se crea automáticamente una variable para almacenar cada uno de los parámetros formales y además se declara dentro de la función una variable local llamada prome. Puede notar que la función promedio tiene una variable c al igual que el programa principal, pero son variables distintas porque están en áreas de memoria separadas. Dentro de la función solo se puede acceder a sus variables locales, por lo tanto, no puede acceder a las variables definidas en el main o en otras funciones. Cada función trabaja con sus propios datos lo que facilita que pueda reutilizarse.

Al invocar a la función desde el programa principal los valores de las variables a, b y c viajan hacia la función, se copian. Esos valores se denominan argumentos de la función y son recibidos y almacenados en las variables definidas automáticamente por los parámetros formales. Desde el programa principal como argumentos de una función se pueden enviar:

- Variables (se enviará el contenido de la misma)
- Constantes
- Operaciones (se resuelve la operación y se envía el resultado)
- Funciones que retornen un valor (al invocar a una función esta si retorna un valor, puede ser utilizado como parámetro de otra función y lo que se envía es el valor retornado por dicha función)