



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

Facultad de Ingeniería de Sistemas Computacionales

Capítulo II

Funciones

Kexy Rodríguez



UNIVERSIDAD
TECNOLOGICA
DE PANAMÁ



Contenido

- Funciones

Facultad de Ingeniería de Sistemas Computacionales

Kexy Rodríguez

Definición de funciones

Matemáticamente una función es una operación que toma uno o más valores llamados argumentos y produce un valor denominado resultado.

Todos los lenguajes de programación tienen funciones incorporadas, intrínsecas o internas.

$$f(x) = (x^2 + 1) + (4 - x)$$

Una función puede tener varios argumentos

$$f(x, y) = (x^2 + 1) + (4 - y)$$

*Las funciones son diseñadas para realizar tareas específicas: toman una lista de valores —llamados argumentos— y **devuelven un único valor**.*

Funciones de Programación

*“En computación, una **subrutina** o **subprograma** (también llamada procedimiento, función o rutina), como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica”.*

“Un bloque de código que puede ser ejecutado como una unidad funcional. Opcionalmente puede recibir valores; se ejecuta y puede devolver un valor. Desde el punto de vista de la organización, podemos decir que una función es algo que permite un cierto orden en una maraña de algoritmos”.

Funciones en C

Conceptos:

En C los módulos se llaman funciones. Por lo general en C los programas se escriben combinando nuevas funciones creadas por el programador con funciones disponibles en la biblioteca estándar de C.

La biblioteca estándar de C contiene una colección de funciones para diferentes propósitos, como cálculos matemáticos, manipulación de cadena, para entrada y salida de datos, y muchas otras operaciones.

Funciones en C

Las funciones en C tienen el siguiente formato:

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



Funciones en C

Cuando se llama a una función se asignan valores a sus parámetros y comienza a ejecutar el cuerpo hasta que se llega al final o se encuentra la instrucción `return`.

```
#include <stdio.h>
```

```
int Sumar(int a, int b)
```

```
{
```

```
    int respuesta;
```

```
    respuesta = a+b;
```

```
    return (respuesta);
```

```
}
```

```
int main()
```

```
{
```

```
    int c;
```

```
    c=Sumar(15, 43);
```

```
    printf("%d",c);
```

```
    return 0;
```

```
}
```

*La instrucción **return** no es una función, se trata de una sentencia que lo que hace es retornar como valor de la función el valor que se le proporciona como argumento.*

Funciones en C

Un aspecto que conviene resaltar es el hecho de que una función también termina su ejecución si llega al final de la misma sin encontrar ninguna sentencia **return**. Ello es posible en toda función de tipo **void**.

```
#include <stdio.h>
```

```
void Sumar(int a, int b)  
{  
    int respuesta;  
    respuesta = a+b;  
    printf("%d",respuesta);  
}
```

```
int main()  
{  
    Sumar(15, 43);  
    return 0;  
}
```


Funciones en C

El **prototipo de una función** es una línea similar a la primera de su declaración: tipo del resultado, seguido del nombre de la función y de la lista de tipos de datos de los parámetros separados por comas y rodeados por paréntesis. Toda función que se invoca debe ir precedida o de su definición o de su prototipo. La definición y el prototipo de la función pueden estar presentes en el mismo archivo.

```
#include <stdio.h>
/* Prototipos */
int Sumar(int a, int b);
/* Función principal */
int main()
{
    int c;
    c=Sumar(15, 43);
    printf("%d",c);
    return 0;
}
/* Definición de funciones */
int Sumar(int a, int b)
{
    int respuesta;
    respuesta = a+b;
    return (respuesta);
}
```



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

