



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Claudia Rodriguez Espino

Asignatura: Fundamentos de Programación

Grupo: 1102

No de Práctica(s): Práctica número 12

Integrante(s): Alejandro Nava Cruz

Semestre: 2018-1

Fecha de entrega: 12 de Noviembre del 2017

Observaciones:

CALIFICACIÓN: _____

Practica 12: Funciones

Objetivo:

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

La sintaxis de una función, al inicializarse es:

Tipo de dato nombre de la función (tipo de dato parámetro);

Para llamar a una función, sólo se debe escribir su nombre y los valores que se le van a enviar. Su sintaxis es la siguiente:

Nombre de la función (Valor);

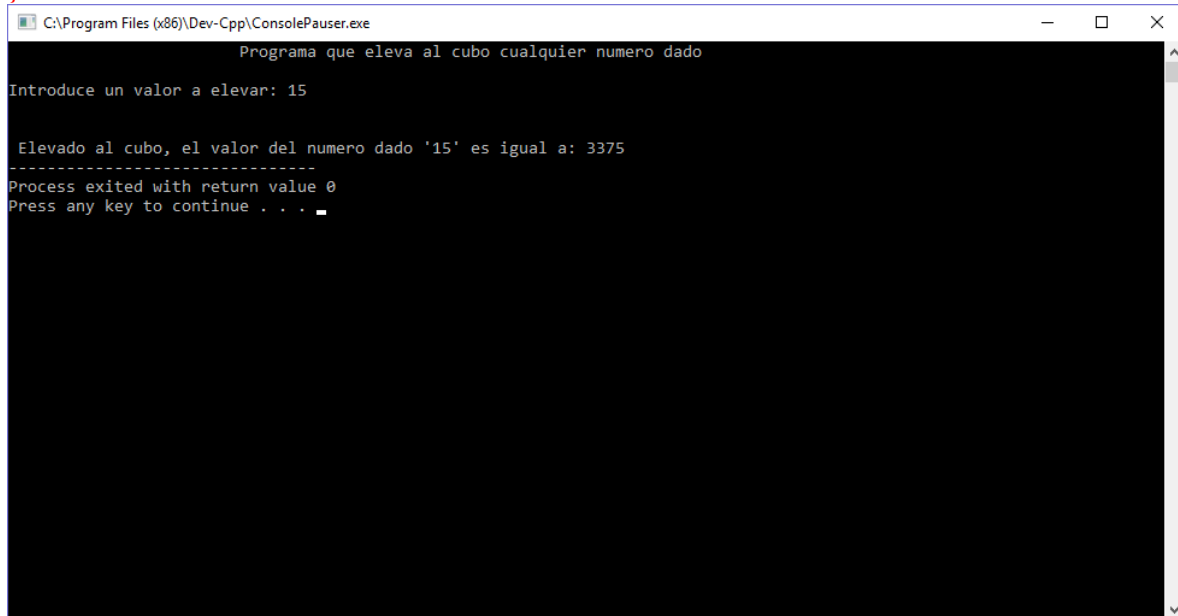
Cabe mencionar, que para que una función regrese una variable, se debe de especificar esto, con el comando `return(variable)`, al final del bloque de instrucciones. La función principal o `main` usará este valor en la instrucción en la que se llamó a la función.

Las funciones se inicializan y se especifican afuera de la función `main`, ya que son independientes de esta.

Actividades para casa/clase

1. Programa que saque la potencia cubica de cualquier numero dado

```
#include<stdio.h>
#include<conio.h>
long cubo(long x);
long in,out;
main()
{
    printf("\t\tPrograma que eleva al cubo cualquier numero dado\n\n");
    printf("Introduce un valor a elevar: ");
    scanf("%ld",&in);
    out=cubo(in);
    printf("\n\n Elevado al cubo, el valor del numero dado '%ld' es igual a: %ld",in,out);
}
long cubo(long x)
{
    long res;
    res=(x*x*x);
    return(res);
}
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Programa que eleva al cubo cualquier numero dado
Introduce un valor a elevar: 15

Elevado al cubo, el valor del numero dado '15' es igual a: 3375
-----
Process exited with return value 0
Press any key to continue . . .
```

2. Programa "Calculadora"

```
#include<stdio.h>
int a,b,c,d,e;
int suma(int a, int b)
{
    d=b+c;
    return(d);
}
int resta(int a,int b)
{
    d=b-c;
    return(d);
}
int multiplicacion(int a, int b)
{
    d=b*c;
    return(d);
}
int division(int a, int b)
{
    d=b/c;
    return (d);
}
main()
{
    printf("\t\t\t\tCalculadora de numeros enteros\nCalculadora, selecciona la operacion que quieres\n");
    printf("hacer.\n1)Suma.\n2)Resta.\n3)Multiplicacion.\n4)Division.\n");
    scanf("%d",&a);
    switch(a)
    {
        case 1:
            printf("Introduce los valores a sumar");
            scanf("%d\n%d",&b,&c);
            e=suma(a,b);
            printf("El resultado de la suma es: %d",d);
            break;
        case 2:
            printf("Introduce los valores a restar");
            scanf("%d\n%d",&b,&c);
            e=resta(a,b);
            printf("El resultado de la resta es: %d",d);
            break;
        case 3:
            printf("Introduce los valores a multiplicar");
            scanf("%d\n%d",&b,&c);
            e=multiplicacion(a,b);
            printf("El resultado de la multiplicacion es: %d",d);
            break;
        case 4:
            printf("Introduce los valores que se van a dividir");
            scanf("%d\n%d",&b,&c);
            e=division(a,b);
            printf("El resultado de la division es: %d",d);
            break;
        default:
            printf("Ese valor no esta permitido ");
            break;
    }
}
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Calculadora de numeros enteros
Calculadora, selecciona la operacion que quieres hacer.
1)Suma.
2)Resta.
3)Multiplicacion.
4)Division.
3
Introduce los valores a multiplicar
7
6
El resultado de la multiplicacion es: 42
-----
Process exited with return value 0
Press any key to continue . . .
```

Conclusiones:

Las funciones y sus principales usos son los de repetir una misma operación varias veces sin necesidad de repetir el código, algo como usar “for, while o do while”, ahorra líneas de código, solo que en algunas ocasiones y con cosas muy sencillas, puede ser más perjudicial y laborioso que eficaz

Bibliografía:

Práctica número 12, Facultad de Ingeniería, en Laboratorios A y B (2017), Sitio Web:
<http://lcp02.fib.unam.mx/>