

### Práctico 2: Funciones y Punteros en C

La teoría general para este práctico puede consultarse en los Capítulos 4 y 5 *Notas de Clase*

1. Ejecute el siguiente programa y diga que hace la función calcula:

```
#include<stdio.h>
int calcula (int, int); /* Prototipo */

void main(){
  int a1,a2, a3;
  printf("Ingrese Numero entero mayor a 0: ");
  scanf("%d", &a1);
  printf("Ingrese Numero entero mayor a 0: ");
  scanf("%d", &a2);
  a3=calcula(a1,a2);
  printf("El resultado es: %d\n",a3);
  getchar();
}

int calcula (int x, int y){
  int cont=0;
  if (x<y) return cont;
  else {
    while (x>=y)
      { cont++ ;
        x -= y; };
      }
  return cont;
}
```

2. Para el siguiente programa, establecer el ámbito de cada uno de los siguientes elementos:

```
#include <stdio.h>
int cube(int y);
int main(void)
{
  int x;
  for (x = 1; x <= 10; x++)
    printf("%d\n", cube(x));
  return 0;
}
int cube(int y)
{ return y * y * y; }
```

- La variable `x` en la función `main`.
- La variable `y` en la función `cube`.
- La función `cube`.
- La función `main`.
- El prototipo de la función `cube`.
- El identificador `y` dentro del prototipo de la función `cube`.

3. Para cada uno de los siguientes puntos, escribir una sentencia que lleve a cabo la tarea indicada. Asumir que las variables `numero1` y `numero2` de tipo `float` se encuentran definidas y que `numero1` ha sido inicializada con `7.3`.

- Definir la variable `fPtr` como un puntero a tipo `float`.
- Asignar la dirección de la variable `numero1` a la variable `fPtr`.
- Imprimir el valor del dato apuntado por `fPtr`.
- Asignar el valor del dato apuntado por `fPtr` a la variable `numero2`.
- Imprimir el valor de `numero2`.
- Imprimir la dirección de la variable `numero1`. Para eso usar el especificador de conversión `%p`.
- Imprimir la dirección almacenada en `fPtr`. Usar el especificador de conversión `%p`. ¿El valor impreso será el valor de la dirección de `numero1`?

4. Implementar una función que solicite el ingreso de 4 notas y que permita obtener el promedio. Mostrar un mensaje “APROBADO”, si su promedio es mayor e igual a 7, en caso contrario visualizar “DESAPROBADO”.

5. Una tienda On-line realiza una promoción 3x2, llevando 3 prendas por el precio de las dos más económicas. Hacer una función que, dados los tres precios de las prendas, retorne la cantidad a pagar.

6. A los operarios de la empresa industrial “INDUSTRI-ROBOT” se les paga sus servicios por horas trabajadas. Si la cantidad de horas trabajadas es mayor a 30 horas se incrementa en un 50% para las horas extras. Implemente una función que realice el cálculo del salario de los operarios pasando por parámetro la cantidad de horas trabajadas. Además debe mostrar un mensaje que según las horas trabajadas muestre el monto de su salario.

7. Escribir una función que tome la hora como tres argumentos enteros (horas, minutos y segundos) y retorne el número de segundos pasados desde las 00:00. Puede usar la función en un programa que calcule el tiempo en segundos entre dos momentos dados.

8. Haga un programa que registre la hora de ingreso y egreso del personal de una fábrica. El programa deberá solamente imprimir un cartel del horario que ingresa o egresa en cada caso. El empleado deberá encontrarse con una pantalla que le dé la opción de Ingresar o Egresar y deberá colocar su código personal (código entero de 4 dígitos). Realice esta funcionalidad para al menos 5 operarios.

Debe tener una función para el ingreso y una para el egreso que imprima en pantalla la leyenda correspondiente al movimiento (INGRESO/EGRESO), el horario y el CODIGO

9. Hacer un programa en C que simule un cajero virtual con un saldo inicial de 3000 pesos (almacenado en una variable definida en el main).

El mismo debe mostrar por pantalla un mensaje de Bienvenida (“Bienvenido al Cajero Virtual en C”).

Además, el usuario deberá seleccionar una opción para poder operar en el mismo. Ejemplo de opciones del cajero:

- <1> Consultar Saldo
- <2> Retirar Dinero
- <3> Depositar Dinero
- <4> Salir

Cada una de las opciones debe ser implementada en una función diferente. En todos los casos el saldo debe ser actualizado según la operación elegida por el usuario.

Cada operación debe mostrar por pantalla un cartel indicando el resultado de la misma.

10. Modifique el ejercicio 11 del práctico 1 de manera tal que realice las mismas funcionalidades pero implementadas a través de módulos.

### Ejercicios Complementarios

1. Completar los siguientes enunciados:

- a. Un módulo en C es llamado un(a) \_\_\_\_\_.
  - b. Una función es invocada con un(a) \_\_\_\_\_.
  - c. Una variable que es conocida sólo dentro de la función en la cual es definida es llamada \_\_\_\_\_.
  - d. La sentencia \_\_\_\_\_ dentro de una función invocada es usada para pasar de vuelta el valor de una expresión a la función invocante.
  - e. La palabra clave \_\_\_\_\_ es usada en el encabezado de una función para indicar que la función no devuelve un valor o para indicar que la función no contiene parámetros.
  - f. El \_\_\_\_\_ de un identificador es la porción del programa en el cual el identificador puede ser usado.
  - g. Las tres maneras de retornar el control desde la función invocada a la invocante son \_\_\_\_\_, \_\_\_\_\_ y \_\_\_\_\_.
  - h. Un(a) \_\_\_\_\_ permite al compilador controlar el número, tipos, y orden de los argumentos pasados a una función.
  - i. Las variables declaradas dentro de un bloque o en la lista de parámetros de una función tienen una permanencia \_\_\_\_\_ a menos que se especifique otra cosa.
  - j. Una variable que no es definida `static` y que se la define fuera de un bloque o función es una variable \_\_\_\_\_.
  - k. Para que una variable local en una función pueda retener su valor entre llamadas debe ser declarada con el especificador \_\_\_\_\_.
  - i. Los cuatro posibles ámbitos de un identificador son \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ y \_\_\_\_\_.
2. Completar cada uno de los siguientes enunciados:
- a. Una variable puntero contiene como valor un(a) \_\_\_\_\_. Tres valores que pueden ser usados para inicializar un puntero son \_\_\_\_\_,

b. \_\_\_\_\_ y \_\_\_\_\_. c. El operador \_\_\_\_\_ retorna el lugar en memoria donde se almacena su operando. d. El operador \_\_\_\_\_ retorna el valor del objeto al cual apunta su operando.

e. Para simular el pasaje por referencia cuando pasamos como parámetro una variable que no es un arreglo, es necesario pasar el/la

\_\_\_\_\_ de la variable a la función.

**3.** Establecer si los siguientes enunciados son verdaderos o falsos:

a. El operador de dirección (&) puede ser aplicado sólo a las constantes.

b. Los punteros de diferentes tipos no pueden ser asignados entre sí sin una operación cast.