

## ***Unidad Nº 3: Actividades sugeridas.***

### ***Preguntas orientadoras***

- 1) Diga en sus palabras que significa ámbito o alcance de una variable.
- 2) ¿Cuál es el peligro de que varias funciones vean o puedan modificar una misma variable?
- 3) ¿Cómo puedo hacer para que varias funciones vean una misma variable sin declararla como global?
- 4) Realice un esquema que relacione el espacio de trabajo (workspace) con distintos proyectos (projects) y sus respectivos archivos fuente, indicando la función que tiene el compilador (compiler) y el linkador (linker) en esas relaciones.
- 5) Analice el efecto del modificador de variable static. Indique que representa este modificador cuando se lo utiliza para funciones.
- 6) En qué situaciones es conveniente pasar un parámetro por valor y en cuales conviene pasarlo por referencia.
- 7) Mencionar una forma equivalente a paso por referencia utilizando paso por valor. ¿Es recomendable utilizar esta forma?
- 8) Llene cada uno de los siguientes espacios en blanco:
  - a) En C un módulo de programa o subprograma se conoce como:\_\_\_\_\_.
  - b) Una función se invoca mediante una \_\_\_\_\_.
  - c) Una variable que es conocida sólo dentro de la función en la cual está definida se conoce como \_\_\_\_\_.
  - d) La palabra reservada \_\_\_\_\_ se utiliza en un encabezado de función para indicar que una función no regresará un valor o para indicar que la función no contiene parámetros.
  - e) La \_\_\_\_\_ o \_\_\_\_\_ le permite al compilador verificar el número, tipos y orden de los argumentos pasados a una función.
  - f) Una función que se llama a sí misma, ya sea directa o indirectamente es una función \_\_\_\_\_.

## Ejercicios

1) Realizar un programa que permita elegir entre calcular el perímetro de una circunferencia, el área de un círculo y el volumen de una esfera. La impresión del menú debe aparecer en una función distinta que la carga de la opción elegida. Ambas definiciones de función, tanto la de impresión del menú como la de lectura de la opción deben aparecer en un único archivo \*.c distinto al main\*. Sus prototipos deben ser definidos en un archivo .h. Finalmente imprimir desde el main la opción seleccionada.

2) Pedir al usuario que ingrese un valor de radio y generar las funciones que calculen lo indicado en el menú del ejercicio 1. Colocar todas las funciones que se parezcan en un mismo archivo \*.c distinto del main.c. Utilizar el #include para incluir las declaraciones de las funciones. Por último mostrar los resultados obtenidos para el radio ingresado.

3) Tipee el siguiente programa en Code Blocks y antes de correrlo comentar cada línea de código indicando que cree que ocurre, luego haga una corrida paso a paso y verifique lo comentado.

```
#include <stdio.h>

void a_local(void);
void b_static(void);
void c_global(void);

int x_global_local = 1; /* */

int main(void)
{
    int x_global_local = 5; /* */
    printf("La variable x_global_local dentro de main es: %d.\n", x_global_local);

    /* */
    {
        int x_global_local = 7;
        printf("La variable x_global_local en el subbloque de main es: %d.\n", x_global_local);/* */
    }

    printf("La variable x_global_local en main es: %d.\n", x_global_local);/* */
    a_local();
    b_static();
    c_global();
    printf("\n\n*****Resultados luego de invocar por segunda vez a las funciones a, b y c*****\n");
    a_local();
    b_static();
    c_global();
    printf("\nLa variable x_global_local en main es: %d.\n", x_global_local);/* */
```

```

    return 0;
}

/* */
void a_local(void)
{
    int x_global_local = 25;
    printf("\nLa variable x_global_local luego de entrar en a_local es: %d.\n", x_global_local);/* */
    x_global_local++;
    printf("La variable x_global_local antes de salir de a_local es: %d.\n", x_global_local);/* */
}

/* */
void b_static(void)
{
    static int x_global_local = 50;
    printf("\nLa variable x_global_local al entrar en b_static es: %d.\n", x_global_local);/* */
    x_global_local++;
    printf("La variable x_global_local antes de salir de b_static es: %d.\n", x_global_local);/* */
}

/* */
void c_global(void)
{
    printf("\nLa variable x_global_local en c_global es: %d.\n", x_global_local);/* */
    x_global_local *= 10;
    printf("La variable x_global_local antes de salir de c_global es: %d.\n", x_global_local);/* */
}

```

4) Encuentre cuales son los errores existentes en los siguientes trozos de programa:

```

a) int func_g(void){printf("Dentro de g\n"); int func_h(void){printf("Dentro de h\n");}}

b) int sum(int x, int y)
    {
        int result;
        result = x + y;
    }

c) void product (void)

    {
        int a, b, c, result;
        printf("Ingrese 3 números: ")
        scanf("%d%d%d", a, b, c);
        result = a * b * c;
        printf("El resultado es %d", result);
        return result;
    }

```

5) Una fábrica de artículos electrónicos contrató a 3 vendedores para la promoción y venta de sus productos. El sueldo de cada vendedor será igual al 10% del monto total de su venta. Realizar un programa que cargando el nombre de cada vendedor y su venta, determine:

- a) El sueldo de cada vendedor.
- b) El monto total de ventas.
- c) El porcentaje que representa cada venta sobre el total de ventas realizadas y las diferencias existentes con los otros vendedores.

6) Realizar las siguientes funciones de conversión en un mismo archivo:

- a) Conversión de decimal a hexadecimal.
- b) Conversión de hexadecimal a decimal.
- c) Conversión de decimal a ASCII.

7) Ingrese un número y realice los siguientes cálculos: raíz cuadrada, logaritmo natural, logaritmo en base 10, elévelo a la 5, realice el cálculo  $e^n$ , calcule el valor absoluto, identifique el entero más próximo por encima y por debajo del número. Ayudita: incluya la librería math.h.

8) Calcular la corriente que circula en una resistencia dados los valores de tensión y resistencia. La función main() se dedica solo a llamar al resto de las funciones que realizan los cálculos y retornan void. Ayudita: La corriente (I) es igual a la razón entre la tensión(V) y la resistencia(R).

