

6. Si una función no regresa un valor, ¿con qué tipo debe ser declarada?
7. ¿Cuál es la diferencia entre una definición de función y un prototipo de función?
8. ¿Qué es una variable local?
9. ¿En qué son especiales las variables locales?

Ejercicios

1. Escriba un encabezado para una función llamada `hazlo()`, que tome tres argumentos de tipo `char` y regrese un tipo `float` al programa que la llama.
2. Escriba un encabezado para una función llamada `imprime_un_número()`, que tome un solo argumento de tipo `int` y no regrese nada al programa que la llama.
3. ¿Qué tipo de valor regresan las siguientes funciones?

- a. `int imprime_error(float num_error);`
- b. `long lee_registro(int num_reg, int longitud);`

4. BUSQUEDA DE ERRORES: ¿Cuál es el error en el siguiente listado?

```
#include <stdio.h>
void print_msg( void );
main()
{
    print_msg( "This is a message to print" );
}

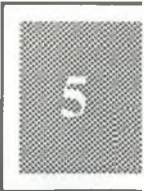
void print_msg( void )
{
    puts( "This is a message to print" );
    return 0;
}
```

5. BUSQUEDA DE ERRORES: ¿Cuál es el error en la siguiente definición de función?

```
int twice(int y);
{
    return (2 * y);
}
```

6. Vuelva a escribir el listado 5.4 de tal forma que sólo necesite un enunciado `return`.

7. Escriba una función que reciba dos números como argumentos y regrese el valor de su producto.
8. Escriba una función que reciba dos números como argumentos. La función debe dividir el primer número entre el segundo. No divida cuando el segundo número sea cero. (Consejo: Use un enunciado `if`.)
9. Escriba una función que llame a las funciones de los ejercicios 7 y 8.
10. Escriba un programa que use una función para encontrar el promedio de cinco valores `tipo float` tecleados por el usuario.
11. Escriba una función recursiva que calcule el valor de 3 a la potencia de otro número. Por ejemplo, si se le pasa 4, la función regresará 81.



Cuestionario

1. ¿Cuál es el valor de índice del primer elemento de un arreglo?
2. ¿Cuál es la diferencia entre un enunciado `for` y un enunciado `while`?
3. ¿Cuál es la diferencia entre un enunciado `while` y un enunciado `do...while`?
4. ¿Es cierto que un enunciado `while` puede ser usado y obtener todavía los mismos resultados que al codificar un enunciado `for`?
5. ¿Qué debe recordarse cuando se anidan enunciados?
6. ¿Puede un enunciado `while` ser anidado en un enunciado `do...while`?

Ejercicios

1. Escriba una declaración para un arreglo que guarde 50 valores de tipo `long`.
2. Muestre un enunciado que asigne el valor de 123.456 al quincuagésimo elemento del arreglo del ejercicio 1.

3. ¿Cuál es el valor de `x` cuando se termina el siguiente enunciado?

```
for( x = 0; x < 100, x++ );
```

4. ¿Cuál es el valor de contador cuando se termina el siguiente enunciado?

```
for( contador = 2; contador < 10; contador +=3 ) ;
```

5. ¿Qué tantas X imprime lo siguiente?

```
for( x = 0; x < 10; x++ )
    for( y = 5; y > 0; y-- )
        puts( "X" );
```

6. Escriba un enunciado `for` para contar de 1 a 100 en incrementos de 3.
7. Escriba un enunciado `while` para contar de 1 a 100 en incrementos de 3.
8. Escriba un enunciado `do...while` para contar de 1 a 100 en incrementos de 3.
9. BUSQUEDA DE ERRORES: ¿Qué está equivocado en el siguiente fragmento de código?

```
record = 0;
while( record < 100 )
{
```

```
printf( "\nRecord %d ", record );  
printf( "\nGetting next number..." );  
}
```

10. BUSQUEDA DE ERRORES: ¿Qué está equivocado en el siguiente fragmento de código? (¡MAXVALUES no es el problema!)

```
for ( counter = 1; counter < MAXVALUES; counter++ );  
printf( "\nCounter = %d", counter );
```

