

## Ejercicios 2.1

1. Indique si los siguientes son nombres válidos de funciones. Si son válidos, indique si son mnemónicos. (Recuerde que el mnemónico de una función transmite una idea acerca del propósito de la función.) Si son nombres inválidos, explique por qué.

poder	densidad	m1234\$	ampnuevo	1234	abcd
total	tangente	valabs	computado	b34a	34ab
voltios\$	a2-b3	while	valMin	seno	\$seno
coseno	velocidad	distancianeta	suma	return	pila

2. Suponga que las siguientes funciones fueron escritas:

`obtenerLargo()`, `obtenerAncho()`, `calcArea()`, `mostrarArea()`

- a. De acuerdo a los nombres de las funciones, ¿qué piensa que puede hacer cada una?
- b. ¿En qué orden piensa que la función `main()` puede ejecutar estas funciones con base en los nombres?

3. Suponga que las siguientes funciones fueron escritas:

`velocidad()`, `distancia()`, `aceleracion()`

De acuerdo a los nombres de las funciones, ¿qué piensa que puede hacer cada una?

4. Determine los nombres para las funciones que hacen lo siguiente:

- a. Encontrar el promedio de un conjunto de números.
- b. Encontrar el área de un rectángulo.
- c. Encontrar el valor mínimo en un conjunto de números.
- d. Encontrar la densidad de una puerta de acero.
- e. Ordenar un conjunto de números del menor al mayor.

5. Así como la palabra clave `int` se utiliza para indicar que una función ha de devolver un número entero, las palabras clave `void`, `char`, `float` y `double` se utilizan para indicar, respectivamente, que una función devolverá sin valor un carácter; un número en punto flotante y un número en doble precisión. Utilizando esta información, escriba las líneas de encabezado para una función de `main()` que no recibirá argumentos pero devolverá.

- a. ningún valor
- b. un carácter
- c. un número en punto flotante
- d. un número en doble precisión

6. a. Utilizando `cout`, escriba un programa en C++ que muestre su nombre en una línea, su dirección en la segunda línea, y la ciudad, estado y código postal en la tercera línea.
- b. Ejecute en una computadora el programa que escribió para el ejercicio 6.a. (NOTA: Debe entender los procedimientos para capturar y ejecutar un programa de C++ en las instalaciones de cómputo que utiliza.)
7. a. Escriba un programa en C++ para desplegar lo siguiente:  
     El coseno de un ángulo  
     es igual a uno dividido entre  
     el seno del ángulo
- b. Compile y ejecute en una computadora el programa que escribió para el ejercicio 7.a.
8. a. ¿Cuántas instrucciones `cout` utilizaría para desplegar lo siguiente?

Grados	Radianes
0	0.0000
90	1.5708
180	3.1416
270	4.7124
360	6.2832

- b. ¿Cuál es el mínimo de instrucciones `cout` que utilizaría para imprimir la tabla en el ejercicio 8.a?
  - c. Escriba un programa completo en C++ para producir la salida mostrada en el ejercicio 8a.
  - d. Ejecute en una computadora el programa que escribió para el ejercicio 8.c.
  9. En respuesta a una secuencia de escape de línea nueva, `cout` coloca el siguiente carácter en pantalla al principio de una nueva línea. Esta manera de colocar el siguiente carácter en realidad representa dos operaciones diferentes. ¿Cuáles son?
  10. a. Suponga que el compilador no es sensible al uso de mayúsculas y minúsculas. ¿Cuáles de estos nombres de unidades de programa son equivalentes?
- |          |          |       |          |      |
|----------|----------|-------|----------|------|
| PROMEDIO | promedio | MODO  | BESSEL   | Modo |
| Total    | bessel   | Temp  | Densidad | TEMP |
| densIDAD | MEDIA    | total | media    | modo |
- b. Vuelva a hacer el ejercicio 10.a pero suponga que el compilador es sensible al uso de mayúsculas y minúsculas.

### Ejercicios para estructurar un proyecto

La mayoría de los proyectos, tanto de programación como de otras áreas, normalmente se puede estructurar en tareas o unidades de actividad más pequeñas. Estas tareas más pequeñas suelen delegarse a diferentes personas; así, cuando todas las tareas estén terminadas e integradas, el proyecto o programa estará terminado. Para los ejercicios del 11 al 16, determine una serie de tareas que, unidas, terminen el proyecto. Tome en cuenta que existen muchas soluciones posibles para cada ejercicio. El único requisito es que cuando se junten las tareas seleccionadas deben completar la tarea requerida.

*NOTA: El propósito de estos ejercicios es que usted analice las diferentes maneras como pueden estructurarse las tareas complejas. Aunque no existe sólo una solución correcta para estos ejercicios, hay soluciones incorrectas y soluciones que son mejores que otras. Una solución incorrecta es una que no lleva a cabo completamente la tarea. Una solución es mejor que otra si establece de manera más clara o fácil lo que se debe hacer.*

11. Le pidieron cablear e instalar las luces en el ático de su casa. Determine la serie de tareas que, en conjunto, han de conseguirlo. (*Sugerencia: La primera tarea sería determinar la ubicación de las lámparas.*)
12. Le encargaron preparar una comida completa para cinco personas a realizarse el siguiente fin de semana. Determine el conjunto de tareas que, reunidas, lo lograrían. (*Sugerencia: Una tarea, no necesariamente la primera, sería comprar los alimentos.*)
13. Usted es un estudiante universitario y después de graduarse piensa ir a una escuela superior para estudiar una maestría en ingeniería eléctrica. Numere los principales objetivos que debe cumplir para lograr esta meta. (*Sugerencia: Una tarea es "Determinar las materias convenientes que debe tomar".*)
14. Le asignaron el trabajo de cultivar un huerto. Determine el conjunto de subtareas que lo conseguirían. (*Sugerencia: Una de estas subtareas sería planear la distribución del huerto.*)
15. Usted es responsable de planear y organizar un viaje familiar al campo este verano. Enumere las subtareas que, en conjunto, conseguirían este objetivo con éxito. (*Sugerencia: Una subtaska sería seleccionar la ubicación en el campo.*)
16.
  - a. Un laboratorio nacional de pruebas médicas desea un sistema nuevo de computadora para analizar los resultados de sus ensayos. El sistema debe ser capaz de procesar los resultados diarios. Además, el laboratorio quiere que el sistema pueda recuperar y producir un reporte impreso de todos los resultados que cumplan con ciertos criterios, por ejemplo, todos los resultados obtenidos por un médico en particular o todos los resultados obtenidos por los hospitales de un estado en particular. Determine tres o cuatro unidades del programa en las cuales el sistema se puede separar. (*Sugerencia: Una posible unidad del programa es "Preparar los resultados diariamente" para crear los reportes de cada día.*)
  - b. Suponga que alguien captura datos incorrectos para el resultado de una prueba específica, lo cual se descubre después de que el sistema introdujo y almacenó los datos. ¿Qué unidad del programa es necesaria para corregir este problema? Discuta por qué puede o no ser necesaria esta unidad del programa, en la mayoría de los sistemas.
  - c. Suponga que existe una unidad del programa que permite al usuario alterar o cambiar datos que fueron introducidos y almacenados de manera incorrecta. Discuta la necesidad de incluir un "rastreo de auditoría" que permitiría una reconstrucción posterior de los cambios efectuados para determinar cuándo se hicieron y quién los realizó.

## Ejercicios 2.2

1. a. ¿Funcionará el siguiente programa?

```
#include <iostream.h>
int main() {cout << "¡Hola mundo!"; return 0;}
```

- b. ¿Por qué no es buen programa el dado en el ejercicio 1.a?

2. Vuelva a escribir los siguientes programas, de manera que cumplan con la buena práctica de programación:

- a. #include <iostream.h>

```
int main(
){
    cout <<
    "El momento ha llegado"
    ; return 0;}
```

- b. #include <iostream.h>

```
int main
(    ){cout << "Newark es una ciudad\n";cout <<
    "En Nueva Jersey\n"; cout <<
    "Tambien es una ciudad\n"
    ; cout << "En Delaware\n"
    ; return 0;}
```

- c. #include <iostream.h>

```
int main() {cout << "Leer un programa\n";cout <<
    "es mas facil\n"
    ; cout << "si utiliza una forma estandar de main\n")
    ; cout
    <<"y cada instruccion se escribe\n";cout
    <<          "en una sola linea\n")
    ; return 0;}
```

- d. #include <iostream.h>

```
int main
(    ){ cout << "Cada programa en C++"
    ; cout
    <<"\ndebe tener una y solo una"
    ;
    cout << "funcion de main"
    ;
    cout <<
    "\n la secuencia de escape de caracteres")
    ; cout <<
```

```
"\npara una linea nueva se puede colocar en cualquier lugar"
; cout
<<"\n dentro del mensaje transmitido a cout"
; return 0;}
```

3. a. Cuando se utiliza en un mensaje, la diagonal inversa alterará el significado del carácter que le sigue inmediatamente. Si quiere imprimir el carácter de diagonal inversa, debe indicar a `cout` que escape de la manera en que normalmente interpreta la diagonal invertida. ¿Cuál carácter utilizará para alterar la manera en la que será interpretado un solo carácter de diagonal inversa?
- b. Utilizando las respuestas del ejercicio 3.a, escriba la secuencia de escape para imprimir una diagonal inversa.
4. a. Una señal ('token' en idioma inglés) en un lenguaje de computadora es cualquier secuencia de caracteres que, como una unidad, sin caracteres que intervengan o espacios en blanco, tiene un significado único. Utilizando esta definición de señal, determine si la secuencia de escape, los nombres de funciones y las palabras clave mostradas en la tabla 1.2 son señales del lenguaje C++.
- b. Discuta si al agregar un espacio en blanco se altera un mensaje. Explique si los mensajes se pueden llamar señales de C++.
- c. Usando la definición de señal, proporcionada en el ejercicio 4.a, determine si la siguiente instrucción es cierta: "Excepto por las señales del lenguaje, C++ ignorará todos los espacios en blanco".

## Ejercicios 2.3

1. Determine los tipos de datos apropiados para los siguientes datos:

- el promedio de cuatro velocidades
- el número de transistores en un circuito
- la longitud del puente Golden Gate
- los números de parte en una máquina
- la distancia de Brooklyn, NY, a Newark, NJ
- los nombres de componentes de un circuito

2. Convierta los siguientes números al formato decimal estándar:

6.34e5    1.95162e2    8.395e1    2.95e-3    4.623e-4

3. Escriba los siguientes números decimales utilizando la notación exponencial:

126.    656.23    3426.95    4893.2    .321    .0123    .006789

4. A continuación se muestran las expresiones algebraicas correctas y las expresiones incorrectas de C++ que corresponden a ellas. Encuentre los errores y escriba las expresiones corregidas en C++.

<i>Álgebra</i>	<i>Expresión en C++</i>
a. $(2)(3)+(4)(5)$	$(2)(3) + (4)(5)$
b. $\frac{6 + 18}{2}$	$6 + 18 / 2$
c. $\frac{4.5}{12.2 - 3.1}$	$4.5 / 12.2 - 3.1$
d. $4.6(3.0 + 14.9)$	$4.6(3.0 + 14.9)$
e. $(12.1 + 18.9)(15.3 - 3.8)$	$(12.1 + 18.9)(15.3 - 3.8)$

5. Determine el valor de las siguientes expresiones en número entero:

- |                         |                         |
|-------------------------|-------------------------|
| a. $3 + 4 * 6$          | f. $20 - 2 / (6 + 3)$   |
| b. $3 * 4 / 6 + 6$      | g. $(20 - 2) / 6 + 3$   |
| c. $2 * 3 / 12 * 8 / 4$ | h. $(20 - 2) / (6 + 3)$ |
| d. $10 * (1 + 7 * 3)$   | i. $50 \% 20$           |
| e. $20 - 2 / 6 + 3$     | j. $(10 + 3) \% 4$      |

6. Determine el valor de las siguientes expresiones en punto flotante:

- a.  $3.0 + 4.0 * 6.0$
- b.  $3.0 * 4.0 / 6.0 + 6.0$
- c.  $2.0 * 3.0 / 12.0 * 8.0 / 4.0$
- d.  $10.0 * (1.0 + 7.0 * 3.0)$
- e.  $20.0 - 2.0 / 6.0 + 3.0$
- f.  $20.0 - 2.0 / (6.0 + 3.0)$
- g.  $(20.0 - 2.0) / 6.0 + 3.0$
- h.  $(20.0 - 2.0) / (6.0 + 3.0)$

7. Suponga que la distancia tiene un valor en número entero de 1, v tiene un valor en número entero de 50, n tiene un valor en número entero de 10, y t tiene un valor en número entero de 5. Evalúe las siguientes expresiones:

- a.  $n / t + 3$
- b.  $v / t + n - 10 * distancia$
- c.  $v - 3 * n + 4 * distancia$
- d.  $distancia / 5$
- e.  $18 / t$
- f.  $-t * n$
- g.  $-v / 20$
- h.  $(v + n) / (t + distancia)$
- i.  $v + n / t + distancia$

8. Utilice los manuales de referencia del sistema para su computadora y determine el código de carácter utilizado por ella.

9. Determine los resultados del siguiente programa:

```
#include <iostream.h>
int main() // un programa que muestra como se trunca //un numero entero
{
    cout << "respuesta1 es el numero entero de " << 9/4 << '\n';
    cout << "respuesta2 es el numero entero de " << 17/3 << '\n';

    return 0;
}
```

10. Determine los resultados del siguiente programa:

```
#include <iostream.h>
int main() // un programa que muestra al operador %
{
    cout << "El residuo de 9 dividido entre 4 es " << 9 % 4 << '\n';
    cout << "El residuo de 17 dividido entre 3 es " << 17 % 3 << '\n';

    return 0;
}
```

11. Escriba un programa en C++ que muestre los resultados de las expresiones  $3.0 * 5.0$ ,  $7.1 * 8.3 - 2.2$  y  $3.2 / (6.1 * 5)$ . Calcule manualmente el valor de estas expresiones para verificar que los valores desplegados sean correctos.
12. Escriba un programa en C++ que despliegue los resultados de las expresiones  $15 / 4$ ,  $15 \% 4$  y  $5 * 3 - (6 * 4)$ . Calcule manualmente el valor de estas expresiones para verificar que los valores desplegados sean correctos.
13.
  - a. Muestre cómo el nombre PULLEY se almacena en una computadora que utiliza código ASCII. Es decir, dibuje una figura similar a la figura 2.7 para las letras de PULLEY.
  - b. Muestre cómo el nombre de PULLEY se almacena en una computadora que utiliza el código EBCDIC.
14.
  - a. Repita el ejercicio 13.a utilizando las letras de su propio apellido.
  - b. Repita el ejercicio 13.b utilizando las letras de su propio apellido.
15. Escriba, compile y ejecute el programa 2.5 en su computadora.
16. Como las computadoras utilizan diferentes representaciones para almacenar los valores de los números enteros, de punto flotante, en doble precisión, y caracteres, explique cómo un programa puede alertar a la computadora sobre los tipos de datos de los diversos valores que estará utilizando.
17. Aunque nos hemos concentrado en operaciones que involucran números enteros y números en punto flotante, C++ permite sumar o restar los caracteres y números enteros. Esto es posible porque C++ siempre convierte un carácter a un valor equivalente en número entero siempre que un carácter se utiliza en una expresión aritmética. De tal manera, los caracteres y números enteros se pueden combinar libremente en estas expresiones. Por ejemplo, si la computadora utiliza código ASCII, la expresión  $'a' + 1$  es igual a  $'b'$ , y  $'z' - 1$  es igual a  $'y'$ . De la misma forma,  $'A' + 1$  es igual a  $'B'$ , y  $'Z' - 1$  es igual a  $'Y'$ . Con estos antecedentes, encuentre los resultados de caracteres de las siguientes expresiones (suponga que todos los caracteres están almacenados utilizando código ASCII).
  - a.  $'m' - 5$
  - b.  $'m' + 5$
  - c.  $'G' + 6$
  - d.  $'G' - 6$
  - e.  $'b' - 'a'$
  - f.  $'g' - 'a' + 1$
  - g.  $'G' - 'A' + 1$

*NOTA: Para el siguiente ejercicio el lector debe haber comprendido los conceptos básicos de almacenaje en la computadora. Específicamente, si desconoce el concepto de byte, consulte la sección 1.6 antes de efectuar el ejercicio.*

18. Aunque el número total de bytes varía de una computadora a otra, los tamaños de memoria de 65,536 a más de varios millones de bytes no son raros. En el lenguaje de computadora, la letra K se utiliza para representar el número 1024, que es 2 elevado a la décima potencia y M se utiliza para representar el número 1,048,576, el cual es 2 elevado a la vigésima potencia. De tal manera, un tamaño de memoria de 640K es en realidad 640 multiplicado por 1024, o 655,360



bytes, y un tamaño de memoria de 4M es en realidad 4 veces 1,048,576, que es 4,194,304 bytes. Utilizando esta información, calcule el número real de bytes en

- a. una memoria que contiene 512K bytes
- b. una memoria que contiene 2 M bytes
- c. una memoria que contiene 8 M bytes
- d. una memoria que contiene 16 M bytes
- e. una memoria que consta de 4 M palabras, donde cada palabra es de 2 bytes
- f. una memoria de 4 M palabras, donde cada palabra es de 4 bytes
- g. un disco flexible que puede almacenar 1.44 M bytes

.

## Ejercicios 2.4

1. Indique si los siguientes nombres de variables son válidos o no. Si son válidos, explique por qué.

prod_a	c1234	abcd	_c3	12345
ampnuevo	vatios	\$total	\$alnuevo	alb2c3d4
9ab6	suma.de	promedio	voltios1	encvoltios

2. Indique si los siguientes nombres de variables son o no válidos. Si son válidos, explique la razón. Señale los nombres válidos de variables que no deben utilizarse porque no transmiten información alguna acerca de la variable.

actual	a234	r2d2	primer_num	cc_al
beto	sandi	c3p0	total	suma
maximo	okay	a	increible	veporel
3suma	for	tot.al	c\$cinco	potencianeta

3. a. Escriba una instrucción de declaración para indicar que la variable de cuenta se utilizará para guardar un número entero.  
b. Escriba una instrucción de declaración para indicar que la variable voltio se utilizará para guardar un número en punto flotante.  
c. Escriba una instrucción de declaración para indicar que la variable potencia se utilizará para guardar un número en doble precisión.  
d. Escriba una instrucción para declarar que la variable de carácter clave se utilizará para guardar un carácter.
4. Escriba las instrucciones declaratorias de las siguientes variables.
  - a. num1, num2, y num3 utilizadas para guardar números enteros
  - b. amps1, amps2, amps3 y amps4 utilizadas para guardar números en punto flotante
  - c. voltios1, voltios2 y voltios3 utilizadas para guardar números en doble precisión
  - d. codigo-a, codigo-b, codigo-c, codigo-d y codigo-e utilizadas para guardar tipos carácter
5. Escriba las instrucciones declaratorias de las siguientes variables:
  - a. primernum y segundonum empleadas para guardar números enteros
  - b. velocidad, aceleración y distancia, utilizadas para guardar números en punto flotante
  - c. empuje utilizada para guardar un número en doble precisión
6. Vuelva a escribir cada una de estas instrucciones de declaración como tres individuales.
  - a. int mes, dia = 30, anho;
  - b. double horas, voltio, potencia = 15.62;
  - c. float precio, cantidad, impuestos;
  - d. char tecla\_entrada, ch, opción = 'f';

7. a. Determine el efecto de cada instrucción en el siguiente programa.

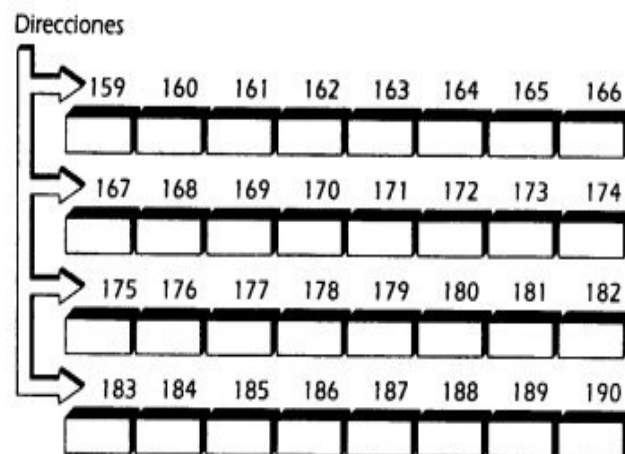
```
#include <iostream.h>

int main()
{
    int num1, num2, total;
    num1 = 25;
    num2 = 30;
    total = num1 + num2;
    cout << "El total de" << num1 << " y "
         << num2 << " es " << total << endl;

    return 0;
}
```

- b. ¿Cuál es la salida que será impresa cuando ejecute el programa mostrado en el ejercicio 7.a?
8. Cada variable tiene tres elementos asociados. ¿Cuáles son?

*NOTA PARA LOS EJERCICIOS DEL 9 AL 11: Suponga que un carácter requiere un byte de espacio de almacenaje, un número entero dos bytes, un número en punto flotante cuatro bytes, un número en doble precisión ocho bytes, y que el espacio de almacenaje se asigna a las variables según el orden en que éstas fueron declaradas (repase la sección 1.6 si no recuerda el concepto de byte).*



**Figura 2.14** Bytes en memoria para los ejercicios 9, 10 y 11.

9. a. Utilice la figura 2.14 y suponga que la variable llamada `rate` está asignada al byte que tiene la dirección de memoria 159. Determine las direcciones correspondientes a cada variable declarada en las siguientes instrucciones. También llene los bytes adecuados con datos de inicialización incluidos en las

instrucciones de declaración (utilice letras para los caracteres, no los códigos de computadora que realmente se almacenarían).

```
float rate;  
char ch1 = 'w', ch2 = 'o', ch3 = 'w', ch4 = '!';  
double taxes;  
int num, count = 0;
```

- b. Repita el ejercicio 9.a, pero sustituya los patrones reales de bytes que usaría una computadora con el código ASCII para guardar los caracteres en las variables ch1, ch2, ch3 y ch4. (*Sugerencia:* Utilice la tabla 2.2.)
10. a. Utilice la figura 2.14 y, suponiendo que la variable de nombre cn1 está asignada al byte en la dirección 159 de memoria, determine las direcciones correspondientes a cada variable declarada en las siguientes instrucciones. Llene los bytes apropiados con los datos de inicialización incluidos en las instrucciones declaratorias (utilice letras para los caracteres y no el código de computadora con el que, en realidad, estarían almacenados).

```
char cn1 = 'a', cn2 = ' ', cn3 = 'b', cn4 = 'u', cn5 = 'n';  
char cn6 = 'c', cn7 = 'h', key = '\\', sch = '\\', inc = 'o';  
char incl = 'f';
```

- b. Repita el ejercicio 10.a, pero sustituya los patrones reales de bytes que usaría una computadora con el código ASCII para guardar los caracteres en cada una de las variables declaradas. (*Sugerencia:* Utilice la tabla 2.2.)
11. Utilice la figura 2.14 y suponga que la variable con nombre millas está asignada al byte en la dirección de memoria 159 y determine las direcciones correspondientes a cada variable declarada en las siguientes instrucciones.

```
float millas;  
int cuenta, num;  
double dist, temp;
```

## Ejercicios 2.5

1. a. Ejecute el programa 2.10 para determinar cuántos bytes reservó la computadora para guardar los tipos de datos carácter y número entero.  
b. Amplíe el programa 2.10 para determinar cuántos bytes utiliza la computadora para números enteros cortos, números enteros largos y los números enteros sin signo.
2. Después de ejecutar el programa escrito para el ejercicio 1.b, use la tabla 2.2 para determinar los números máximo y mínimo que puede guardar en las variables en número entero, número entero corto y número entero largo en su computadora.
3. El programa 2.10 en realidad no guardó un valor en las variables `ch` y `num1`. ¿Por qué no fue necesario esto?
4. a. Amplíe el programa 2.10 para determinar cuántos bytes utiliza su computadora para guardar números en punto flotante y en doble precisión.  
b. Aunque no existe una clase de datos largos y flotantes, los números en doble precisión a veces se consideran como equivalentes a una forma larga para números en punto flotante. ¿A qué se debe esto? ¿Apoya esta afirmación la salida del programa escrito para el ejercicio 4.a?

## Ejercicios 2.6

NOTA: En cada uno de estos ejercicios exponemos un problema de programación. Primero lea el planteamiento y después conteste las preguntas relacionadas con el problema. No escriba un programa para resolver el problema; simplemente conteste las preguntas que siguen a la especificación del programa.

1. Suponga que debe escribir un programa en C++ para calcular la resistencia total de un circuito en serie. En tal circuito, la resistencia total es la suma de todos los valores individuales de las resistencias. Suponga que el circuito consiste en una cantidad de resistores de 56 ohmios, 33 ohmios, y 15 ohmios.
  - a. ¿Cuántas salidas requiere para este problema de programación?
  - b. ¿Cuántos datos de entrada tiene este problema?
  - c. Determine un algoritmo para convertir los elementos de insumo en elementos de salida. Suponga que la cantidad de resistores de 56 ohmios es  $m$ , la cantidad de resistores de 33 ohmios es  $n$ , la cantidad de resistores de 15 ohmios es  $p$ .
  - d. Pruebe el algoritmo escrito para la parte 1.c utilizando la siguiente muestra de datos:  $m = 17$ ,  $n = 24$ , y  $p = 12$ .

2. Suponga que el programa ha de escribirse para calcular el valor de la *distancia*, en millas, conforme la relación:

$$\text{distancia} = \text{velocidad\_de\_viaje} * \text{tiempo\_transcurrido}$$

- a. ¿Cuántas salidas requiere este problema de programación?
- b. ¿Cuántos datos de insumo requiere el problema?
- c. Determine un algoritmo para convertir los elementos de insumo en elementos de salida.
- d. Pruebe el algoritmo escrito para la parte 2.c empleando los siguientes datos de muestra: la *velocidad\_de\_viaje* es 55 millas por hora y el *tiempo transcurrido* es 2.5 horas.
- e. ¿Cómo debe modificar el algoritmo que determinó en la parte 2.c si el tiempo transcurrido le fuera dado en minutos en vez de horas?

3. Suponga que debe escribir un programa para determinar el valor de *ergios* con las siguientes relaciones:

$$\text{ergios} = \text{fergios} * \sqrt{\text{lergios}}$$

$$\text{lergios} = 2 * \pi * e^{\mu}$$

- a. ¿Cuántas salidas requiere para este problema de programación?
- b. ¿Cuántos datos de insumo requiere este problema?
- c. Determine un algoritmo para convertir los elementos de insumo en elementos de salida.
- d. Pruebe el algoritmo escrito para la parte 3.c utilizando los siguientes datos de muestra:  $\text{fergios} = 14.65$ ,  $\pi = 3.1416$ ,  $\mu = 1.672$  y  $e = 2.7818$ .

4. Suponga que ha de escribir un programa para desplegar las siguientes especificaciones:

Amplificación de voltaje: 35  
Salida de potencia: 2.5 vatios  
Ancho de banda: 15KHz

- ¿Cuántas líneas de salida requiere para este problema de programación?
  - ¿Cuántos datos de insumo tiene este problema?
  - Determine un algoritmo para convertir los elementos de insumo en elementos de salida.
5. Escriba un programa en C++ para determinar la distancia recorrida por un automóvil en 10 segundos, suponiendo que inicialmente viajaba a 60 millas por hora y el conductor aplica el freno para reducir la velocidad a razón de 12 millas/seg<sup>2</sup>. En este caso,  $distancia = s - (1/2)dt^2$ , donde  $s$  es la velocidad inicial del automóvil,  $d$  es la desaceleración y  $t$  es el tiempo transcurrido.
- ¿Cuántas salidas se requieren en este problema de programación?
  - ¿Cuántos datos de insumo tiene este problema?
  - Determine un algoritmo para convertir los elementos de insumo en elementos de salida.
  - Pruebe el algoritmo escrito en la parte 5.c empleando los datos dados en el problema.
6. Considere el siguiente problema de programación: En 1627, la isla de Manhattan fue vendida a los holandeses por aproximadamente \$24 dólares. Si los ingresos de esa venta se hubieran depositado en un banco holandés que paga 5% de interés compuesto anual, ¿cuál sería el saldo principal al final del año 2002? Necesitará una pantalla similar a la siguiente: El Saldo al 31 de diciembre de 2002 es: xxxxxx, donde xxxxxx es la cantidad calculada por el programa.
- ¿Cuántas salidas requiere para este problema de programación?
  - ¿Cuántos datos de insumo tiene este problema?
  - Determine un algoritmo para convertir los datos de insumo en elementos de salida.
  - Pruebe el algoritmo escrito para la parte 6.c empleando los datos proporcionados en el planteamiento del problema.
7. Escriba un programa que calcule y despliegue los voltajes de salida para dos circuitos eléctricos y la suma de los dos voltajes. La salida de voltaje para el primer circuito está dada por la ecuación  $(150)V/0.38f$  y el voltaje de salida para el segundo circuito está dado por la ecuación

$$\frac{230V}{\sqrt{56^2 + (0.98f)^2}}$$

donde  $V$  es el voltaje introducido al circuito y  $f$  es la frecuencia en hertzios.

- ¿Cuántas salidas requiere para este problema de programación?
- ¿Cuántos datos de entrada tiene este problema?
- Determine un algoritmo para convertir los elementos de entrada en elementos de salida.
- Pruebe el algoritmo escrito para la parte 7.c utilizando los siguientes datos de muestra: El primer circuito opera con voltaje de entrada de 1.2 voltios a una

frecuencia de 144 hertzios y el segundo circuito lo hace con un voltaje de entrada de 2.3 voltios a 100 hertzios.

8. Considere el siguiente problema de programación: La fórmula para la desviación estándar normal,  $z$ , utilizada en aplicaciones de estadística es

$$z = \frac{X - \mu}{\sigma}$$

donde  $m$  se refiere al valor medio y  $\sigma$  es la desviación estándar. Utilizando esta fórmula, escriba un programa que calcule y despliegue el valor de la desviación estándar normal cuando  $X = 85.3$ ,  $\mu = 80$  y  $\sigma = 4$ .

- ¿Cuántas salidas requiere para este problema de programación?
  - ¿Cuántos datos de entrada tiene este problema?
  - Determine un algoritmo para convertir los elementos de entrada en elementos de salida.
  - Pruebe el algoritmo escrito para la parte 8.c utilizando los datos proporcionados en el problema.
9. La ecuación de una curva normal (en forma de campana) utilizada en aplicaciones estadísticas es

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Utilizando esta ecuación, suponga que debe escribir un programa en C++ que calcule el valor de  $y$ .

- ¿Cuántas salidas se requieren para este problema de programación?
- ¿Cuántos datos de entrada tiene este problema?
- Determine un algoritmo para convertir los elementos de entrada en elementos de salida.
- Pruebe el algoritmo escrito para la parte 9.c, suponiendo que  $\mu = 90$ ,  $\sigma = 4$ ,  $x = 80$  y  $\pi = 3.1416$ .



## Ejercicios 2.7

1. a. Modifique el programa 2.12 para calcular la velocidad de un auto con una frecuencia del radar de  $2.00000035 \times 10^{10} \text{ seg}^{-1}$ .  
b. Compile y ejecute en su computadora el programa escrito para el ejercicio 1.a.
2. a. Modifique el programa 2.12 para determinar la frecuencia que registraría un auto que viaja a 55 millas por hora. El programa debe producir el siguiente resultado:

La frecuencia devuelta que corresponde a 55 millas por hora es \_\_\_\_\_  
donde el subrayado será reemplazado por el valor real calculado por el programa.

- b. Compile y ejecute en su computadora el programa escrito para el ejercicio 2.a. Asegúrese de hacer un cálculo a mano para que pueda verificar los resultados producidos por su programa.
- c. Después de verificar los resultados del programa escrito en el ejercicio 2.a, modifique el programa para calcular la frecuencia producida por un auto que viaja a 75 millas por hora.
3. a. Modifique el programa 2.13 para calcular y desplegar el número total de líneas necesarias para conectar directamente 1000 teléfonos individuales entre sí.  
b. Compile y ejecute en su computadora el programa escrito para el ejercicio 3.a.
4. a. Modifique el programa 2.13 para que una variable nueva, numfin (el número adicional de suscriptores que se conectarán a la red existente), se inicialice en 10. Haga los cambios necesarios en el programa para que produzca un resultado similar al del programa 2.13.  
b. Compile y ejecute en su computadora el programa escrito para el ejercicio 4.a. Revise que el resultado producido por el programa coincida con el mostrado en el libro.
5. a. Diseñe, escriba, compile y ejecute un programa en C++ para convertir la temperatura en grados Fahrenheit a grados Celsius. La ecuación para esta conversión es  
$$\text{Celsius} = 5.0/9.0 (\text{Fahrenheit} - 32.0).$$
  
Haga que su programa convierta y despliegue la temperatura en grados Celsius que corresponde a 98.6 grados Fahrenheit. El programa debe producir el siguiente despliegue:  
Para una temperatura Fahrenheit de \_\_\_\_\_ grados,  
la temperatura Celsius equivalente es de \_\_\_\_\_ grados.  
donde los valores apropiados han de ser insertados por su programa en lugar de las líneas subrayadas.  
b. Revise a mano los valores calculados por el programa. Después de haber verificado que su programa funciona correctamente, modifíquelo para convertir 86.5 grados Fahrenheit a su valor equivalente en grados Celsius.

6. a. Escriba, compile y ejecute un programa C++ para calcular la resistencia de un circuito en serie que consta de 12 resistores de 56 ohmios, 20 de 39 ohmios, 32 de 27 ohmios y 27 de 15 ohmios. Recuerde que la resistencia de un circuito en serie es la suma de todas las resistencias individuales. El programa debe producir el siguiente despliegue:

La resistencia total, en ohmios, es xxxx

dónde xxx será reemplazado por el valor real de resistencia calculado por el programa.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para calcular la resistencia de un circuito en serie que consta de ningún resistor de 56 ohmios, 17 resistores de 39 ohmios, 19 resistores de 27 ohmios y 42 resistores de 15 ohmios.

7. a. Diseñe, escriba, compile y ejecute un programa en C++ para calcular el tiempo necesario para efectuar un viaje de 183.67 millas. La ecuación para calcular el tiempo transcurrido es

$$\text{tiempo transcurrido} = \text{distancia total} / \text{velocidad promedio}$$

Suponga que la velocidad promedio durante el viaje es de 58 millas por hora.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para calcular el tiempo necesario para efectuar un viaje de 372 millas a una velocidad promedio de 67 millas/hora.

8. a. Diseñe, escriba, compile y ejecute un programa en C++ para sumar los números del 1 al 100. La fórmula para calcular esta suma es

$$\text{suma} = (n/2)(2*a + (n - 1)d)$$

dónde  $n$  = número de términos que debe sumar,  $a$  = el primer número y  $d$  = la diferencia entre cada número.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para calcular la suma de los números enteros de 100 a 1000.

NOTA: Los ejercicios 9, 10 y 11 requieren elevar un número a una potencia. Esto se puede lograr con la función de potencia `pow()` en el C++. Por ejemplo, la instrucción `pow(2.0, 5.0)`; eleva el número 2.0 a la quinta potencia, y la instrucción `pow(num1, num2)`; eleva la variable `num1` a la potencia `num2`. Para utilizar la función de potencia oloque un comando `#include <math.h>` del preprocesador en una línea individual después del comando `#include <iostream.h>` o incluya la instrucción de declaración `double pow();` en las instrucciones declaratorias variables utilizadas en su programa. La función de potencia se explicará con más detalle en la sección 3.3.

9. a La ley de enfriamiento de Newton establece que cuando un objeto con una temperatura inicial  $T$  se introduce en una sustancia a una temperatura de  $A$ , aquél llegará a la temperatura  $TFIN$  en  $t$  minutos, de acuerdo con la siguiente fórmula:

$$TFIN = (T - A) e^{-kt} + A$$

En esta fórmula,  $e$  es el número irracional 2.71828 redondeado a 5 cifras decimales, normalmente conocido como número de Euler, y  $k$  es un coeficiente

térmico, el cual depende del material que se está enfriando. Usando esta fórmula, escriba, compile y ejecute un programa en C++ que determine la temperatura de un objeto que ha estado 20 minutos en un vaso de agua cuya temperatura es de 60 grados. Suponga que la temperatura inicial del objeto era de 150 grados y que tiene una constante térmica de 0.0367.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para determinar la temperatura del objeto después de haber estado 10 minutos en un vaso de agua cuya temperatura sea de 50 grados.

10. a. La ganancia de voltaje de un amplificador está dado por la siguiente fórmula:

$$\text{ganancia de voltaje} = [23 / (23^2 + (0.5f)^2)^{1/2}]^n$$

donde  $f$  es la frecuencia, en hertzios, y  $n$  es el número de etapas en el amplificador. Utilizando esta fórmula, escriba, compile y ejecute un programa en C++ que determine el valor de ganancia del voltaje para un amplificador de cuatro etapas que opera a una frecuencia de 120 hertzios. El programa debe producir el siguiente despliegue:

En una frecuencia de xxxxx hertzios, el aumento de voltaje es yyyyyy

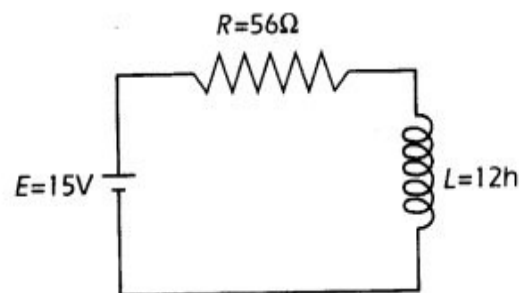
donde xxxxx será reemplazado por la frecuencia y yyyyyy por la ganancia de voltaje.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para determinar la ganancia de voltaje en un amplificador de 12 etapas que opera a una frecuencia de 9500 hertzios.

11. a. La corriente eléctrica,  $i$ , en amperios, que fluye a través del circuito mostrado en la figura 2.18 está indicada por la siguiente ecuación:

$$i = \frac{E(1 - e^{-(R/L)t})}{R}$$

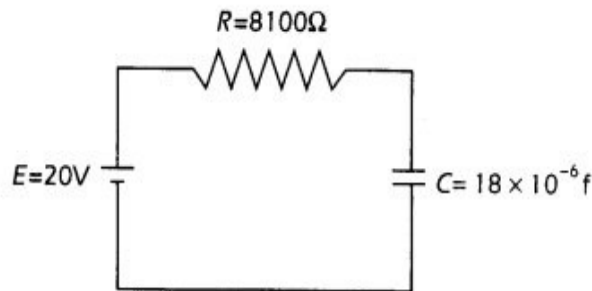
donde  $E$  es el voltaje de la batería en voltios,  $R$  es el valor del resistor en ohmios,  $L$  es el valor del inductor en henrios,  $t$  es el tiempo en segundos después de cerrar el interruptor y  $e$  es el número de Euler, el cual es 2.718, correcto a tres cifras decimales. Utilizando esta fórmula, escriba, compile y ejecute un programa en C++ para determinar el flujo de corriente en el circuito mostrado en la figura 2.18 cuando  $t$  es 0.12 segundos.



- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para determinar la corriente en 0.12 segundos cuando  $E$  es 25 voltios,  $R$  es 33 ohmios, y  $L$  es 15 henrios.
12. a. La corriente eléctrica,  $i$ , en amperios, que fluye a través del circuito mostrado en la figura 2.19 está dada por la siguiente ecuación:

$$i = \frac{(E)e^{-t/RC}}{R}$$

donde  $E$  es el voltaje de la batería en voltios,  $R$  es el valor del resistor en ohmios,  $C$  es el valor del capacitor en faradios,  $t$  es el tiempo en segundos después de cerrar el interruptor y  $e$  es el número de Euler, el cual es 2.718, correcto a tres cifras decimales. Utilizando esta fórmula, escriba, compile y ejecute un programa en C++ para determinar el voltaje a través del capacitor mostrado en la figura 2.19, cuando  $t$  es de 0.31 segundos.

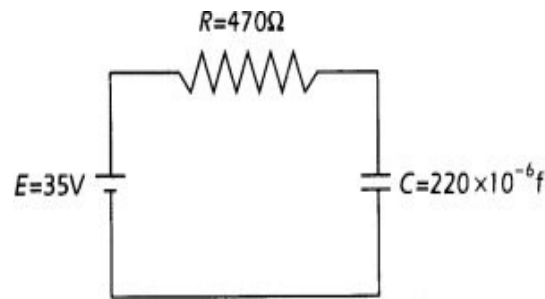


**Figura 2.19** Un circuito RC en serie.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para determinar la corriente en 0.85 segundos cuando  $E$  es 25 voltios,  $R$  es 220 ohmios y  $C$  es 0.00039 faradios.
13. a. El voltaje,  $V$ , en voltios, a través del capacitador  $C$ , mostrado en la figura 2.20, está dado por la siguiente ecuación:

$$V = \frac{E[1 - e^{-t/RC}]}{R}$$

donde  $E$  es el voltaje de la batería en voltios,  $R$  es el valor del resistor en ohmios,  $C$  es el valor del capacitor en faradios,  $t$  es el tiempo en segundos después de cerrar el interruptor y  $e$  es el número de Euler, el cual es 2.718, correcto a tres cifras decimales. Utilizando esta fórmula, escriba, compile y ejecute un programa en C++ para determinar el voltaje a través del capacitor mostrado en la figura 2.20, donde  $t$  es 0.42 segundos.



**Figura 2.20** Un circuito RC en serie.

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para determinar la corriente en 0.85 segundos cuando  $E$  es 25 voltios,  $R$  es 220 ohmios y  $C$  es 0.00039 faradios.

14. a. El conjunto de ecuaciones lineales de

$$a_{11}X_1 + a_{12}X_2 = c_1$$

$$a_{21}X_1 + a_{22}X_2 = c_2$$

se puede solucionar utilizando la regla de Cramer:

$$X_1 = \frac{c_1 a_{22} - c_2 a_{12}}{a_{11} a_{22} - a_{12} a_{21}}$$

$$X_2 = \frac{c_2 a_{11} - c_1 a_{21}}{a_{11} a_{22} - a_{12} a_{21}}$$

Utilizando estas ecuaciones, escriba, compile y ejecute un programa en C++ para encontrar los valores de  $X_1$  y  $X_2$  que satisfacen las siguientes ecuaciones

$$3X_1 + 4X_2 = 40$$

$$5X_1 + 2X_2 = 34$$

- b. Revise manualmente los valores calculados por el programa. Después de haber verificado que el programa funciona correctamente, modifíquelo para solucionar el siguiente par de ecuaciones:

$$3X_1 + 12.5X_2 = 22.5$$

$$4.2X_1 - 6.3X_2 = 30$$