

- 5.21 Utilice técnicas similares a las desarrolladas en los ejercicios 5.19 y 5.20 para producir un programa que grafique una amplia gama de formas
- 5.22 Escriba segmentos de programa que lleven a cabo cada uno de ellos lo siguiente:
- Calcule la parte entera del cociente cuando el entero a se divide por el entero b .
 - Calcule el residuo entero cuando el entero a es dividido por el entero b .
 - Utilice las porciones de programa desarrolladas en a) y en b) para describir una función que introduzca un entero entre 1 y 32767 y lo imprima como una serie de dígitos, estando separado cada par de dígitos por dos espacios. Por ejemplo el entero 4562 deberá ser impreso como

4 5 6 2

- 5.23 Escriba una función que obtenga el tiempo como tres argumentos enteros (para horas, minutos y segundos) y regrese el número de segundos desde la última vez que el reloj "llegó a las 12". Utilice esta función para calcular la cantidad de tiempo en segundos entre dos horas, cuando ambas estén dentro de un ciclo de 12 horas del reloj.
- 5.24 Ponga en marcha las siguientes funciones enteras:
- La función `celsius` que regresa el equivalente Celsius de una temperatura en Fahrenheit.
 - La función `fahrenheit` que regresa el equivalente en Fahrenheit de una temperatura en Celsius.
 - Utilice ambas funciones para escribir un programa que imprima gráficas mostrando los equivalentes Fahrenheit de todas las temperaturas Celsius desde 0 hasta 100 grados, y los equivalentes Celsius de todas las temperaturas Fahrenheit entre 32 y 212 grados. Imprima las salidas en un formato tabular nítido, que minimice el número de líneas de salida manteniéndose legible.
- 5.25 Escriba una función que regrese el más pequeño de tres números de punto flotante.
- 5.26 Un número entero se dice que se trata de un *número perfecto* si sus factores, incluyendo a 1 (pero excluyendo en el número mismo), suman igual que el número. Por ejemplo, 6 es un número perfecto porque $6 = 1+2+3$. Escriba una función `perfect` que determine si el parámetro `number` es un número perfecto. Utilice esta función en un programa que determine e imprima todos los números perfectos entre 1 y 1000. Imprima los factores de cada número perfecto para confirmar que el número de verdad es perfecto. Ponga en acción la potencia de su computadora para probar números más grandes que 1000.
- 5.27 Se dice que un entero es *primo* si es divisible sólo entre 1 y sí mismo. Por ejemplo, 2, 3, 5, y 7 son primos, pero 4, 6, 8 y 9 no lo son.
- Escriba una función que determine si un número es primo.
 - Utilice esta función en un programa que determine e imprima todos los números primos entre 1 y 10,000. ¿Cuántos de estos 10,000 números tendrá que probar verdaderamente antes de estar seguro de que se han encontrado todos los números primos?
 - Inicialmente pudiera pensar que $n/2$ es el límite superior para el cual debe usted probar para ver si un número es primo, pero sólo necesita llegar hasta la raíz cuadrada de n . ¿Por qué? Vuelva a escribir el programa, y ejecútelo de ambas formas. Estime la mejoría en rendimiento.
- 5.28 Escriba una función que tome un valor entero y regrese el número con sus dígitos invertidos. Por ejemplo, dado el número 7631, la función debería regresar 1367.
- 5.29 El *máximo común divisor* de dos enteros es el entero más grande que divide de forma uniforme cada uno de los dos números. Escriba una función `gcd` que regrese el máximo común divisor de dos enteros.
- 5.30 Escriba una función `qualityPoints` que introduzca el promedio de un alumno y regrese 4 si el promedio es entre 90 - 100, 3 si el promedio es entre 80 - 89, 2 si el promedio es entre 70 - 79, 1 si el promedio está entre 60 - 69 o si el promedio es menor de 60.