

Fundamentos de Programación

Convocatoria de Septiembre. Curso 2012/2013

Tiempo: 2.5 horas

1. (2.5 puntos) Deseamos implementar un módulo que, dada una secuencia de dígitos de longitud arbitraria, calcule un dígito de control. Este dígito se utiliza para la detección de errores en la transmisión de dicha secuencia. En concreto, debes realizar un módulo que reciba una secuencia de caracteres (cada uno de ellos será un dígito del '0' al '9') y que devuelva un carácter con el dígito de control (también del '0' al '9'). El algoritmo utilizado será el siguiente: sumar los dígitos de las posiciones impares (SI), sumar los dígitos de las posiciones pares (SP) y calcular $S = SP + 3 * SI$. El dígito de control será la diferencia entre la decena inmediatamente superior a S y S. Si esa diferencia es diez entonces el dígito de control será cero. Por ejemplo:

- Secuencia de dígitos 1738603229867591209: (SI=1+3+6+3+2+8+7+9+2+9=50, SP=7+8+0+2+9+6+5+1+0=38, S=38+3*50=188). DC=190-188=2
- Secuencia de dígitos 19473819: (SI=1+4+3+1=9, SP=9+7+8+9=33, S=33+3*9=60). DC=70-60=10 \Rightarrow DC=0

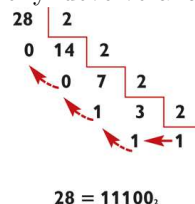
2. (2.5 puntos) Se desea implementar una clase para representar un número complejo. Un complejo se define como un par ordenado de números reales (a, b) , donde a representa la parte real y b la parte imaginaria.

- a) Defina los atributos de la clase y el constructor o constructores que estime oportunos.
- b) Implemente un método para la suma de dos complejos definida como $(a, b) + (c, d) = (a + c, b + d)$
- c) Implemente un método para el producto de dos complejos definido como $(a, b) \cdot (c, d) = (ac - bd, ad + bc)$
- d) Implemente un método para determinar si dos complejos son iguales: $(a, b) = (c, d)$ si y solo si $a = c$ y $b = d$.
- e) Implemente un método que determine la distancia euclídea entre dos números complejos (a, b) y (c, d) . La distancia se define como $\sqrt{(c - a)^2 + (d - b)^2}$.
- f) Implemente un módulo para que dado un vector de complejos y un número complejo, devuelva la posición en el vector del complejo más cercano.

3. (2 puntos) Implementar una función **RECURSIVA** que dado un número en base decimal obtenga su representación en binario como un vector de enteros. La función recibirá un entero y devolverá un vector.

Recordad que para pasar a binario, hay que ir dividiendo por 2 el entero hasta llegar a un cociente menor que dos. El bit más significativo es el último cociente y los demás dígitos son los restos obtenidos de las divisiones, desde el último hasta el primero.

Por ejemplo, si el entero es 28, la función debe devolver el vector de enteros {1,1,1,0,0}



4. (3 puntos) **Sudoku** es un juego muy popular que consiste en rellenar una cuadrícula de 9×9 celdas que está dividida en subcuadrículas de 3×3 (denominadas *regiones*) con cifras del 1 al 9. Un sudoku se considera resuelto si verifica que:

- En cada fila aparecen todos los números del 1 al 9 (sin repetir)
- En cada columna aparecen todos los números del 1 al 9 (sin repetir)
- En cada región aparecen todos los números del 1 al 9 (sin repetir)

Realizar un programa que lea todos los elementos de un sudoku y determine si está resuelto o no. Se valorará la correcta modularización del programa. Un ejemplo de sudoku resuelto es el siguiente:

1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	4	5	6	7	8	9	1
5	6	7	8	9	1	2	3	4
8	9	1	2	3	4	5	6	7
3	4	5	6	7	8	9	1	2
6	7	8	9	1	2	3	4	5
9	1	2	3	4	5	6	7	8