

Instituto de Tecnología ORT II
Programación III
Laboratorio de Programación III
Segundo Cuatrimestre 2007

Práctica 1
Estructuras de Datos Estáticas

- 1) Dado un vector de n elementos, diseñe e implemente un algoritmo que realice la búsqueda secuencial de un elemento x . Estime la cantidad de comparaciones cuando el elemento se encuentra en el vector y cuando no existe en el mismo.
- 2) Dado un vector de n elementos, diseñe e implemente un algoritmo que realice la búsqueda binaria de un elemento x . Estime la cantidad de comparaciones cuando el elemento se encuentra en el vector y cuando no existe en el mismo.
- 3) Diseñe e implemente un algoritmo que traduzca un número de su representación binaria a su representación decimal.
- 4) Diseñe e implemente un algoritmo que compruebe si un número binario tiene la misma cantidad de ceros y de unos.
- 5) Diseñe e implemente un algoritmo que reciba como argumento una matriz cuadrada cuyos coeficientes son números enteros y calcule la suma de los cuadrados de los coeficientes de dicha matriz.
- 6) Considere la estructura de un *cuadrado mágico*, el cual consiste en una matriz cuadrada cuyos coeficientes son números enteros, donde la suma de los números de cada fila, columna y diagonal es la misma. Diseñe e implemente un algoritmo que reciba como argumento una matriz de números enteros y determine si es un cuadrado mágico.
- 7) Dado un vector de números enteros, diseñe e implemente un algoritmo que determine si alguno de los elementos del vector coincide con la suma de todos los que le preceden.
- 8) Dado un vector de números enteros, diseñe e implemente un algoritmo que determine si alguno de los elementos del vector coincide con la suma de todos los que le suceden.
- 9) Dado un vector de números enteros y un entero n , diseñe e implemente un algoritmo que determine si existen en el vector dos números cuya suma sea n .
- 10) Diseñe e implemente un algoritmo que determine la cantidad de términos de la serie armónica que se necesitan para satisfacer la desigualdad:
 $1 + 1/2 + 1/3 + \dots + 1/n > \text{límite}$, donde *límite* es un valor dado.
- 11) Diseñe e implemente un algoritmo *CalcularLongitud* que reciba como argumento una cadena de caracteres c y determine la longitud de la misma.

- 12) Diseñe e implemente un algoritmo *CompararCadenas* que reciba como argumentos dos cadenas de caracteres c_1 y c_2 y determine si son iguales.
- 13) Diseñe e implemente un algoritmo *ConcatenarCadenas* que reciba como argumentos dos cadenas de caracteres c_1 y c_2 y permita obtener una cadena de caracteres c_3 resultado de concatenar c_1 y c_2 .
- 14) Diseñe e implemente un algoritmo *ObtenerSubcadena* que reciba como argumentos una cadena de caracteres c_1 y un número entero n mayor a 0 , y permita obtener una cadena de caracteres c_2 con los n primeros caracteres de c_1 . El algoritmo debe controlar que sea posible obtener los n caracteres de acuerdo a la longitud de c_1 .
- 15) Diseñe e implemente un algoritmo *InvertirCadena* que reciba como argumento una cadena de caracteres c_1 y permita obtener una cadena de caracteres c_2 resultado de invertir c_1 .
- 16) Diseñe e implemente un algoritmo *IntersecarCadenas* que reciba como argumentos dos cadenas de caracteres c_1 y c_2 y permita obtener una cadena de caracteres c_3 que contenga los caracteres de c_1 que existen en c_2 . El algoritmo debe controlar que si existen caracteres repetidos en c_1 que existen en c_2 , estén una única vez en c_3 .
- 17) Diseñe e implemente un algoritmo *BurbujeoCadenas* que reciba como argumento una cadena de caracteres c y utilice el método de burbujeo para ordenar en forma ascendente los caracteres de c .
- 18) Diseñe e implemente un algoritmo que reciba como argumentos una cadena de caracteres c y un carácter c_1 y retorne un carácter c_2 que indique:
 - *N* si c_1 no existe en c .
 - *I* si la parte izquierda antes de c_1 tiene la misma longitud y los mismos caracteres que la parte derecha luego de c_1 .
 - *D* si la parte izquierda antes de c_1 y la parte derecha luego de c_1 son distintas.
- 19) Diseñe e implemente un algoritmo que reciba como argumentos dos cadenas de caracteres c_1 y c_2 , busque la primera aparición de c_1 en c_2 y retorne el índice de c_2 donde comienza c_1 .
- 20) Diseñe e implemente un algoritmo que reciba como argumento una cadena de caracteres c y determine si es palíndroma.