

Practico 0 – Ejercicios de Repaso Algoritmos y E. D. 1

Ejercicio 1

Implemente un método que imprima en consola la suma de todos los impares entre 1 y 50.

Ejercicio 2

Implemente un método que reciba un int e imprima los dígitos de forma separada. Asuma que el número ingresado es positivo.

Pruebelo con los siguientes valores: 5, 1523, 20, 100000.

Ejercicio 3

Diseñar una rutina recursiva que liste los números del 1 al n. ¿Podría lograr que los liste en orden inverso?

Ejercicio 4

Implementar una función recursiva que calcule el factorial de un número natural N.

Ejercicio 5

Dado un arreglo de números enteros, implementar utilizando recursión:

- a) Implementar una función que reciba el arreglo y retorne su mínimo elemento.
- b) Implementar una función que reciba el arreglo y retorne la posición del máximo elemento.
- c) Usando la función definida en la parte b) implemente una función recursiva que ordene un arreglo.

Ejercicio 6

Cree la clase Funcionario con atributos nombre y sueldo. Incluir métodos de acceso y modificación. Agregar dos métodos de clase para prueba:

--- ejemplo1: crea un funcionario con datos a ingresar por el operador y lo muestra

--- ejemplo2: crea dos funcionarios (usando el ejemplo1) e indica además cuál gana más.

Ejercicio 7

Suponga que al siguiente array de enteros se lo está ordenando mediante algún algoritmo de ordenación.

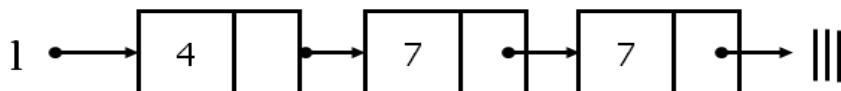
35	7	67	52	31	28
7	35	67	52	31	28
7	28	67	52	31	35
7	28	31	52	67	35

- a) ¿Mediante cuál de los algoritmos vistos en el curso se está ordenando el array? Justifique su respuesta.
- b) Termine de correr el algoritmo mostrando cómo queda el array luego de cada pasada.
- c) Explique cómo funciona el algoritmo *MergeSort* (Ordenación por Intercalación) y aplíquelo al array del ejemplo mostrando el resultado obtenido en cada una de las pasadas.

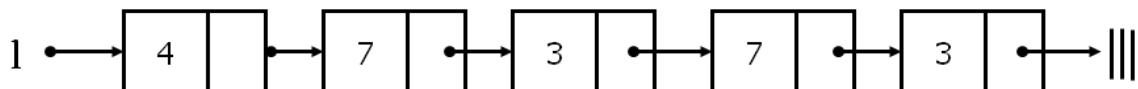
Ejercicio 8

- a) **(5 puntos)** Implemente una función que dada una lista simplemente encadenada de enteros, retorne la cantidad de elementos (nodos) de la lista.
- b) **(15 puntos)** Implemente una función **RECURSIVA** que dada una lista simplemente encadenada de enteros, un entero a y un entero b retorne la lista resultado de insertar b luego de todas las ocurrencias de a en la lista.

Ejemplo: Para la siguiente lista, $a = 7$ y $b = 3$



Retorna:



Ejercicio 9

Desarrolle, aplicando una técnica de dividir para conquistar, una función que, recibiendo un array de **enteros positivos**, un entero izq y un entero der, retorne la posición sobre el array de la primera ocurrencia del menor múltiplo de 3 y la cantidad de veces que se repite entre las posiciones izq y der del array.

Si no hubiera ningún múltiplo de 3 entre las posiciones izq y der se retornará repeticiones 0 y posición 0.

Para ello deberá definir:

- Cuál es un problema fácil y como se resuelve en este caso
- Cómo se dividen los problemas difíciles
- Como se recuerdan y combinan las soluciones a los subproblemas para producir la solución del problema original.

La firma sugerida es void posicionMenorMultiploDe3(int v[],int izq, int der,Entero posicion, Entero repeticiones). Donde Entero es una clase auxiliar para poder pasar por referencia los valores.

```
class Entero {  
    public int valor;  
    public Entero(){  
        this.valor = 0;  
    }  
}
```

Por ejemplo posicionMenorMultiploDe3 (2,4) retorna posicion 3 repeticiones 1
posicionMenorMultiploDe3 (1,7) retorna posicion 1 repeticiones 2.

VALOR	5	3	4	9	5	2	6	3	1	5
POSICION	0	1	2	3	4	5	6	7	8	9