



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Práctica 14

Datos generales:

Nombre de la Tarea	Recursión
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	III. Mecanismos de programación complementarios.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	21-mayo-2025
Fecha de entrega con extensión y penalización	22-mayo-2025 7:30 am
Fecha de elaboración	20-mayo-2025

Objetivo de la tarea	Familiarizarse con el uso de la recursión.
Tiempo aproximado de realización	3 horas
Introducción	La recursividad es una poderosa herramienta que permite simplificar muchos programas que por su naturaleza son recursivos y que su solución iterativa puede resultar más compleja.

Referencias que debe consultar el alumno (si se requieren):

Referencia 1:

1.Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.

Referencia 2:

2.Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.

Actividades que debe realizar el alumno:

Actividad inicial:

Lea primero toda la práctica. No inicie a programar sin leer todo cuidadosamente primero. Recuerde que debe generar el reporte en formato IDC.

Actividad 1:

Primero genere la **Introducción**.

Actividad 2:

Utilizando la actividad 3 de la práctica 9 (el enunciado de encriptar y des-encriptar cadenas). Haga un programa recursivo que lea un archivo de texto (usando los programas que ya tenemos que leen un archivo de texto plano) en un arreglo y encripte cada posición del arreglo (el cual es una cadena, equivalente a una línea de texto de un archivo) y lo guarde en otro arreglo paralelo, el cual deberá ser grabado en otro archivo de texto. Posteriormente haga el programa que des-encripte siguiendo la misma lógica, grabando el resultado en un archivo.

Haga el programa (actividad 2, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Actividad 3:

Crea un método recursivo en Java que compruebe si un número es binario. Un número binario está formado únicamente por ceros y unos. Por ejemplo, un número binario es: 101001; un número no binario es 123001. El parámetro proporcionado como argumento por el usuario es un número de tipo de dato long.

Haga el programa (actividad 4, la cual es el **desarrollo** del programa, junto con la captura de pantalla del programa funcionando).



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Actividad 4:

Construya una función en Java que convierta un número decimal en una cadena que represente el valor del número en hexadecimal (base 16).

Recordatorio: El cambio de base se realiza mediante divisiones sucesivas por 16 en las cuales los restos determinan los dígitos hexadecimales del número según la siguiente correspondencia:

Resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dígito	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Por ejemplo:

$$\begin{array}{r} 65029 \div 16 = 4064 \text{ R } 5 \\ 4064 \div 16 = 254 \text{ R } 0 \\ 254 \div 16 = 15 \text{ R } 14 \\ 15 \div 16 = 1 \text{ R } 15 \end{array}$$

5 0 14 15
↓ ↓ ↓ ↓
5 0 E F → FE05

$$65029_{10} = FE05_{16}$$

Haga el programa (actividad 5, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Actividad 5:

Hacer un programa recursivo en Java que convierta un número decimal en binario.

Consiste en un método parecido a la factorización en números primos. Es relativamente fácil dividir cualquier número entre 2. Este método consiste básicamente en divisiones sucesivas. Dependiendo de si el número es par o impar, colocaremos un cero o un uno en la columna de la derecha.

Si es impar, le restaremos uno y seguiremos dividiendo entre dos (y podremos un 1 en el lado derecho como anteriormente expongo), hasta llegar al resultado final que debe ser siempre 1.

Después, sólo nos queda tomar los resultados de la columna de la derecha y ordenar los dígitos de abajo para arriba, y tendremos nuestro número convertido en binario.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Ejemplo:

150|0

75|1*

37|1

18|0

9|1

4|0

2|0

1|1

El resultado para 150 en base decimal es: 10010110 en base binaria.

*Aquí ponemos 1 al lado derecho y restamos 1 de 75 para poder seguir dividiéndolo entre 2, el resultado lo ponemos debajo, y así sucesivamente.

Haga el programa (actividad 7, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Actividad 6:

Pruebe el funcionamiento del programa de las actividades con todo y sus capturas de pantalla.

Actividad 7:

Realice la sección de **Código agregado** (diagrama de clases UML).

Actividad 8:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Actividad 9:

Finalmente haga las **Conclusiones**.

Actividad 10:

Enviar en <http://ingsoftware.reduaz.mx/moodle>

Archivo anexo que se requiere para esta tarea (opcional):

Dudas o comentarios: a7donso@gmail.com