



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Práctica 22

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) | II. Recursión. |
| Docente que imparte la materia | Aldonso Becerra Sánchez |
| Fecha de entrega para los alumnos | 13-octubre-2021 |
| Fecha de entrega con extensión y penalización | 14-octubre-2021 |
| Fecha de elaboración | 13-octubre-2020 |

| | |
|----------------------------------|--|
| Objetivo de la tarea | Familiarizarse con el uso de la recursión. |
| Tiempo aproximado de realización | 2 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:

Generar el reporte en formato IDC.

Actividad 1:

Primero genere la **Introducción**.

Actividad 2:

Realizar un método recursivo que haga el proceso de la multiplicación entera, es decir:

$$a * b = a + a + \overset{(b \text{ veces})}{\dots} + a$$

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

Actividad 3:

Realizar un programa recursivo que calcule la serie

$$1 - x^1/1! + x^3/3! - x^5/5! + x^7/7! \dots x^n/n! + n/m + n$$

Haga el programa (actividad 3, la cual es el **desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

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). A continuación, generalice la función para convertir un número decimal en un número en base B (con $B < 10$).

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:

65029 \div 16 = 4064 R 5
 4064 \div 16 = 254 R 0
 254 \div 16 = 14 R 14 (E)
 14 \div 16 = 0 R 14 (E)
 Reading remainders from bottom to top: 5, 0, E, F \rightarrow FE05

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

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

Actividad 5:

Para calcular el máximo común divisor de dos números enteros puedo aplicar el algoritmo de Euclides, que consiste en ir restando el más pequeño del más grande hasta que queden dos números iguales, que serán el máximo común divisor de los dos números. Por ejemplo, si comenzamos con el par de números 412 y 184, tendríamos:

| | | | | | | | | | | | | |
|-----|-----|-----|-----|----|----|----|----|----|----|----|---|---|
| 412 | 228 | 44 | 44 | 44 | 44 | 44 | 36 | 28 | 20 | 12 | 8 | 4 |
| 184 | 184 | 184 | 140 | 96 | 52 | 8 | 8 | 8 | 8 | 8 | 4 | 4 |

Haga el método en Java que permita hacer ese proceso.

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

Actividad 6:

Hacer un programa recursivo 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.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

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.

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 6, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Actividad 7:

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



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Actividad 8:

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

Actividad 9:

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

Actividad 10:

Finalmente haga las **Conclusiones**.

Actividad 11:

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

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

Dudas o comentarios: a7donso@gmail.com