

GUÍA DE TRABAJO PRÁCTICO

CURSADA 2024 – 1ER CUATRIMESTRE

Práctica I: Recursividad

1. Construir una función recursiva que retorne verdadero si una cadena de caracteres pasada como parámetro es un palíndromo. Ej. “neuquen” ya que se lee igual de atrás hacia adelante.

Ejemplos:

palindromo(“neuquen”) => verdadero

palindromo(“palindromo”) => falso

2. Dados dos números enteros m y n, construir una función recursiva que devuelva el producto de ambos, calculando el mismo como sumas sucesivas. Esto es:

$m * n = m + m + \dots + m$, n veces.

Ejemplos:

producto(5, 3) => 15

producto (10, 4) => 40

3. Generar un algoritmo recursivo que calcule el k-ésimo número de la serie de Fibonacci.

Ejemplos:

terminoSeridFibonacci (1) => 1

terminoSeridFibonacci (5) => 8

4. Dados los números enteros m y n, construir una función recursiva que devuelva el cociente de ambos, calculando el mismo mediante restas sucesivas. Se deberá tener en cuenta que en el caso de que la división no sea exacta, se devolverán hasta 5 cifras decimales (si es necesario).

Ejemplos:

division (10,2) => 5

division (22,3) => 7,33333

5. Generar un algoritmo recursivo que le ponga los “.” de los miles a un String de números.

Ejemplos:

agregarSeparadorMiles (“123456”) => “123.456”

agregarSeparadorMiles (“12345678”) => “12.345.678”

6. Se conoce que la mafia china es muy organizada y protege mucho a sus miembros, cuando deciden asistir a una reunión se dispone de una cantidad de chinos que asisten, y ellos se ubican de forma que al mirarlos frontalmente generan cierto respeto

y temor. A continuación, se tiene una serie de posibles reuniones y su nivel y la apariencia que se tiene del grupo que va a la reunión vistos frontalmente:

Nivel reunión	Vista frontal de la delegación
1	(-.)
2	(-.(-.)-.)
3	(-.(-.(-.)-.)-.)
4	(-.(-.(-.(-.)-.)-.)-.)

Considerando esta descripción, diseñar los procesos recursivos que permitan mostrar la apariencia del grupo de chinos que asisten a una reunión de nivel n.

7. Se tiene una cadena que representa una onda digital de señales L (Low) y H (High). Se pide obtener recursivamente la onda que representa utilizando “_” y “|”.

Ejemplos:

ondaDigital (“LHLHL”) => “_ | _ | _ | _”

ondaDigital (“HHHHLLLLHHHHLLHH”) => “- - - - | _ _ _ _ | - - - - | _ _ | - - -”

8. Definir una función recursiva que dado un conjunto devuelva una lista con los subconjuntos del mismo tales que la suma de los elementos de cada subconjunto sumen una cantidad dada. Por ejemplo: Dado el conjunto $A = \{10, 3, 1, 7, 4, 2\}$. La lista de los conjuntos que sumen 7 sería: $R = [\{3, 4\}, \{1, 4, 2\}, \{7\}]$

Ejemplos:

subconjuntosQueSumanN ($\{10, 3, 1, 7, 4, 2\}, 7$) => $\{\{3, 4\}, \{1, 4, 2\}, \{7\}\}$

subconjuntosQueSumanN ($\{10, 3, 1, 7, 4, 2\}, 10$) => $\{\{10\}, \{3, 7\}, \{3, 1, 4, 2\}, \{1, 7, 2\}\}$

9. Escribir una función recursiva que implemente el método que se describe para saber si un número es divisible por 7. Se separa la primera cifra de la derecha, se la multiplica por 2, y se resta este producto de lo que queda a la izquierda y así sucesivamente, hasta que el resultado obtenido sea un número menor a 70. El número original será múltiplo de 7 si el resultado da cero o múltiplo de 7.

Ejemplos:

32291 -> $1 \times 2 = 2$.

3229 - 2 =

3227 -> $7 \times 2 = 14$.

322 - 14 =

308 -> $8 \times 2 = 16$

30 - 16 =

14 -> Múltiplo de 7

divisiblePor7 (32291) => verdadero

110 -> $0 \times 2 = 0$

11 - 0 =

11 -> No múltiplo de 7

divisiblePor7 (110) => falso

10. Se dice que **N** es un número explosivo cuando éste explota en varios fragmentos más chicos que él, dada una bomba **B**. Si se tiene que N es el número y B la bomba, tales



que N es mayor que B , se puede hacer que N explote en dos números $N_1 = N / B$ (división entera) y $N_2 = N - (N / B)$.

Pero B es una bomba que produce una reacción en cadena, si N_1 o N_2 son mayores que B , estos también explotan con la regla anterior, hasta que se encuentre que el número no es mayor que B ; entonces se dice que ya no se puede explotar el número.

Escribe una función recursiva que retorne una lista con los pedazos del número N , dado que se tiene la bomba B .

Ejemplos:

Número: 10

Bomba: 3

explosion(10, 3) => [3 2 1 1 3]

Número: 20

Bomba: 5

explosion(20, 5) => [4 3 2 2 1 1 1 5]

IMPORTANTE !!!

- Los prácticos podrán ser resueltos en grupos de 4 (**cuatro**) a 7 (**siete**) alumnos. Condición indispensable para que los TPs sean evaluados.
- Todos los prácticos son obligatorios.
- Los prácticos deben ser entregados en tiempo y forma en las fechas que los docentes estipulen (serán determinadas según la cursada y avisadas mediante la plataforma virtual).
- Los prácticos NO se recuperan por lo que si son entregados en condiciones incorrectas de funcionamiento, fuera de término o fuera de formalidad (abajo descripto) se consideran desaprobados.
- Se deben presentar todos los ejercicios del tp. Se deben tener como mínimo un 60% de los ejercicios correctos para aprobar la práctica, con 2 de los 3 últimos ejercicios funcionando correctamente. Se evalúan tanto los errores de lógica como los de ingreso de datos. Deben controlar los ingresos de datos, tales como blancos, ingreso de números correctos, rangos válidos, etc.
- Los prácticos deben presentarse con la formalidad:
 - Carátula Impresa (Fecha de Entrega, Integrantes de Grupo, Título del Práctico a Entregar cuando la entrega es presencial). Si es virtual deben notificar por mail que el TP está disponible en el GIT para su corrección.
 - Se tomará asistencia en el momento de entrega del TP.
 - El código fuente debe estar disponible en GIT HUB, con acceso a los docentes. Si por algún motivo no se puede acceder o faltan archivos, o sus "**include**" están incorrectos y no se los puede compilar el TP se considera fuera de forma y por lo tanto será desaprobado.

FECHA DE ENTREGA: 08/04/2024 (hasta las 17 hs).

El lenguaje de programación será "C". (Usar el foro para consultas sobre el TP)

Profesor a Cargo de la Asignatura: Mario Perello.

Trabajos Prácticos a Cargo de: Claudia Reinaudi, José Racker, Pablo Chale y Mariano Goldman.