IMPORTANTE: En todos los ejercicios implemente en pseudocódigo y luego en Java, y realice trazas para verificación

En todos los ejercicios implemente en pseudocódigo y luego en Java, y realice trazas para verificación.

**1.** Determinar a través de trazas cual es el cálculo realizado por siguiente función:

public static int func(int n)

{ //precondicion: n es un entero positivo

int aux;

if (n==0)

aux = 0;

else

aux = n + func(n-1);

return aux;

}

1. Escribir una función iterativa que realice la misma función y repetir las trazas.

**2.** Determinar mediante una traza cuál es la salida que arroja la siguiente función recursiva,

además de deducir cual es el cálculo realizado.

public static int algo (int n)

{ //precondicion: n es un entero positivo

int aux;

if (n==1)

aux = 1;

else

aux= algo(n-1) + 1;

return aux;

}

**3.** Determinar mediante una traza cuál es la salida que arroja el siguiente módulo recursivo.

public static void main (String [] args)

{ //averiguar que hace mistery

String numero;

numero=”123456789";

int j;

j = numero.length()-1;

mistery(j , numero);

}

public static void mistery(int p, String a)

{

if (p>0)

{

System.out.println(a.charAt(p));

mistery(p-1,a);

}

}

**4.** Diseñar e implementar un algoritmo recursivo para verificar si todos los dígitos de un número son iguales (por ejemplo si el número es 111 debe retornar true, si el número es 12111 debe retornar false).

**5.** Diseñar e implementar un algoritmo recursivo que lea una secuencia de dígitos **(en cada llamada recursiva debe leer un número)** y cuente cuántos son múltiplos de 3 hasta que se ingrese el valor 0.

**6.** Diseñar e implementar un algoritmo recursivo para obtener el número mayor de un arreglo numérico.

**7.** Diseñar e implementar un algoritmo recursivo para calcular y mostrar la suma de cada una de

las filas de una matriz de enteros.

**Utilice dos módulos recursivos, uno para recorrer la matriz y mostrar el resultado, y otro para el cálculo de la suma.**

**8.** Diseñar e implementar un algoritmo recursivo que retorne true si una oración es un palíndromo.

Un palíndromo es una oración capicúa (no se contabiliza las mayúsculas y minúsculas ni los espacios en blanco):

Ejemplos de palíndromos: “NEUQUEN”; “Somos o no somos”; “Yo hago yoga hoy” “Anita lava la tina”; “Sé verlas al revés”.

**9.** Diseñar e implementar un algoritmo recursivo que dados dos números (divisor y dividendo) realice la división por restas sucesivas.

*El algoritmo de restas sucesivas para realizar la división entera de dos números positivos consiste en ir restando al dividendo el divisor hasta obtener un resultado menor que el divisor, que será el resto de la división. El número de restas efectuado será el cociente.*

*Ej: 56 div 15 = 3 y de resto 11*

*56 – 15 = 41 🡪 Como 41 es mayor a 15, sigo restando*

*41 – 15 = 26 🡪 Como 26 es mayor a 15, sigo restando*

*26 – 15 = 11 🡪 11 es el resto (porque es menor a 15) y como hemos hecho 3 restas, el cociente es 3.*

**10**. Diseñar e implementar un algoritmo recursivo que lea una secuencia de caracteres hasta que se ingrese el punto y retorne una frase formada por los caracteres en el orden leído.

**11**. Diseñar e implementar un algoritmo recursivo que dado un arreglo de números realice dos sumas separadas, por un lado los números en las posiciones pares y por el otro las posiciones impares.

**12**. Diseñar e implementar un algoritmo recursivo que dado un arreglo de caracteres y un caracter indique si se encuentra el caracter en el arreglo dado.

**13**. Diseñar e implementar un algoritmo recursivo que obtenga el número mayor de una matriz numérica.

**14**. Dada una matriz de enteros implementar un algoritmo recursivo que calcule la suma de los elementos de cada columna y vaya mostrando el valor sumado para cada columna.

**15**. Diseñar e implementar un algoritmo recursivo que dado una cadena de caracteres cuente cuantas vocales contiene.

**16**. Dado un arreglo de TDA producto realizado en el práctico 3, diseñar e implementar un algoritmo recursivo que permita retornar la cantidad de productos con precio unitario menor a un valor dado.

**17.** Escriba un planteo recursivo para dibujar una media pirámide de dígitos como se muestra en la siguiente figura.  La cantidad de renglones depende de un n ingresado por el usuario (en el caso del ejemplo es 9)

1

21

321

4321

54321

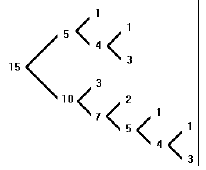
654321

7654321

87654321

987654321

1. Dados dos números N y B, tales que B<N, podemos hacer que N explote usando a B como bomba. Cuando N explota se parte en dos números N1=(N div B) y N2=N-(N div B). Pero B produce una reacción en cadena: si N1 (y/o N2) es mayor que B, también explota y se parte nuevamente en dos pedazos, según el criterio anterior. Esto se repite hasta que todos los pedazos resultantes a partir de N sean menores o iguales que B.

Ej,: Si N=15, y la bomba B=3, el número N se parte inicialmente en dos: 15 div 3, y 15-(15 div 3), es decir, 5 y 10. Como ambos son mayores que la bomba, deben estallar en dos. El proceso se repite según lo muestra la figura.

Escriba un procedimiento recursivo **Explotar,** que dado un número N y un número *bomba* B, imprima todos los pedazos que quedan al explotar N usando B. Ej.: En el caso que muestra la figura, deberá imprimirse 1,1,3,3,2,1,1,3

1. Escriba un planteo y procedimiento recursivo para imprimir una pirámide de dígitos como se muestra en la siguiente figura. Utilice un procedimiento recursivo para generar cada una de las filas de la pirámide. Realice una traza para verificar su correcto funcionamiento. La cantidad de renglones depende de un n ingresado por el usuario (en el caso del ejemplo es 8)

  1

          121

        12321

      1234321

    123454321

  12345654321

                 1234567654321

        123456787654321