Análisis Numérico (752)
Cátedra: Javier Ignacio Garcia Fronti
Curso: Julio Fabris

# Práctica N°3

## *Creación de funciones*

1) Escriba un programa – función cuya salida sea el elemento n-ésimo de la sucesión de Fibonacci. La sucesión de Fibonacci tiene como sus dos primeros elementos el 0 y el 1. Los restantes términos se calculan como la suma de los dos anteriores.

2) Escriba un programa (función) que determine si un número entero positivo (argumento) es par o impar. Recuerde que puede hacerlo comprobando si el número es divisible exactamente por 2. La salida deberá ser 1 si es par y 0 si no es par. Luego, utilizando ese programa como esclavo, escriba una nueva función que cuente los número pares e impares de una matriz de tamaño arbitrario que contiene elementos enteros positivos. La salida de este programa será una matriz cuyo primer elemento sea la cantidad de valores pares en A y el segundo elemento la cantidad de valores impares en A.

4) Escriba un programa (función) que calcule el número *pi* mediante simulación de 'lluvia' al azar, usando *n* observaciones (argumento de la función).

El proceso es el siguiente:

Se generan puntos al azar (x, y) con x, y pertenecientes a [0, 1] y se verifica si pertenecen o no al interior de un cuarto de círculo de radio uno (Es decir, se comprueba si x^2 + y^2 < 1). La proporción de puntos en el interior del círculo converge en probabilidad al área de ese cuarto de círculo a medida que se usan más puntos. Nota: El área de un cuarto de círculo de radio 1 es A=pi/4.

5) Escriba una función que devuelva el mínimo elemento del vector *x* (argumento).

6) Escriba una función que, dado un vector, calcule la varianza, pudiendo indicar como argumento de la función, si se busca una varianza muestral o poblacional.

7) Escriba una función que se aplique sobre un escalar *x* (asumimos entero y positivo) y determine si x es primo o no. Un número primo es aquel que sólo es divisible por 1 y por sí mismo. Para saberlo deberá dividir el número por todos los otros entre 1 y x y ver si surge algún resto cero. Si el número es primo el programa retorna un valor 1. Si no lo es retorna un valor 0.

8) Escriba una función en R que aplique el programa anterior como esclavo para determinar si los elementos de una matriz o vector ***y*** de enteros positivos son primos. El programa deberá retornar un vector o matriz de igual dimensión que ***y***, pero con 1´s donde los elementos correspondientes de ***y*** son primos y 0´s donde los elementos de ***y*** son no primos.

9) Escriba un programa tipo función que genere una matriz A de dimensión *m x n*. Dicha matriz deberá tener como elementos números primos entre 0 y 101.

10) Escriba una función en R tal que tome a un vector x como argumento de entrada y devuelva un vector y cuyos elementos surgen de ordenar x de menor a mayor mediante el siguiente procedimiento ("Método de la Burbuja" o "bubble sort"): Se recorre todo el vector x comparando cada elemento con el anterior. Si están en orden incorrecto se permutan y se continúa avanzando, comparando y si es necesario, permutando. Una vez que se llega al final de x se vuelve a comenzar. El proceso termina cuando, ante un recorrido completo en x no se realiza ninguna permutación

12) Escriba un programa con formato de función que realice lo siguiente: Encontrar el número más pequeño en una matriz dada y reportarlo, así como su posición en la matriz. Si se repite, reportar todas las posiciones en que se encuentra. El input deberá ser una matriz arbitraria de *m x n* y el output, el escalar correspondiente al valor mínimo, así como el vector con la posición del número encontrado (todas las posiciones, si hubiera más de una).

13) En una lista de números enteros consecutivos desde "a" hasta "b" encontrar aquellos que son divisibles por "c". Reportar un vector con los números que cumplan la condición. Los argumentos deberán ser un vector y el escalar por el cual se quiere dividir.

14) Dada una matriz dato genere una función que retorne la imagen espejada de la original respecto del eje vertical.

15) Escribir un programa que recorra los elementos de un vector v y encuentre los que son divisibles por un número dato x.
Reporte una copia del vector donde los elementos divisibles se reemplacen por "div".
Ejemplo: v=(1:9); w=[1 2 DIV 4 5 DIV 6 7 DIV]

16) Programe una función que detecte si un número entero positivo es o no un cuadrado perfecto: cuadrado(x). Utilizando la función anterior como "esclavo" programe una función "maestro" que tenga como argumento una matriz cualquiera de orden *m x n*.

17) Esta función detecta(A) debe detectar los cuadrados perfectos de la matriz de entrada y devolver una matriz similar pero con dichos cuadrados perfectos reemplazados por *CP*'.

18) Dado un vector de números *v* (dato) eliminar los elementos repetidos generando un nuevo vector *w*. con los elementos de este vector construir un nuevo vector *u* que contenga todos los productos posibles formados por 2 elementos distintos de *w*. Reportar el vector *u*.

19) Dada una matriz A de n x 2, donde cada fila representa las coordenadas (x , y) de un punto en el plano. Determinar cuáles de dichos puntos se encuentran dentro de la zona determinada por los ejes cartesianos y la recta y = a - b x , con a y b positivos dados por el vector "param". Reportar las filas que representan puntos dentro de la zona.

20) Escriba un programa que detecte en un vector dato los números que llamaremos "contiguos", que resultan del producto de dos enteros seguidos, por ejemplo 6 = 2 x 3 ó 56 = 7 x 8. El reporte del programa será una matriz de n x 3 donde en la primera columna sea la posición del número contiguo en el vector y las otras dos sean los factores en que se descompone el contiguo.

Sugerencia: pruebe con los enteros cercanos a la raíz del número.

21) Programar una función cuyo desempeño sea generar un vector con elementos enteros positivos aleatorios entre 0 y 100, llamado "a", de longitud "n" (argumento). Con los elementos de dicho vector "a" generar otro vector "b" cuyo primer elemento sea 1 y los elementos siguientes se calculan sumando y restando alternativamente los números del vector a. La cantidad de elementos del vector b deberá ser, entonces n+1.