

Algoritmos

No 1 - Given an array and an element, find the index of that element in the array. If it does not exist in the array, return -1. If there are multiple occurrences of that element in the array, return the index of first occurrence

Example

Test Case 1: [1,2,1,3,2,4,2,3,3,1,3,3], 3
Expected Output: 3

Test Case 2:[1,2,3,4], 5
Expected Output: -1

INICIO

```
VARIABLE matriz[] entero, numero entero, cantidad entero

MENSAJE "ingrese la cantidad de datos que contendrá la matriz"

RECIBIR (cantidad)

PARA i de 0 hasta (cantidad-1)
    |   matriz[i] = aleatorio(10)
FIN PARA

MENSAJE "ingrese el número a buscar"

RECIBIR (numero)

VARIABLE indice entero

indice = -1

PARA i de 0 hasta (cantidad-1)
    |   SI (indice = -1)
        |       SI (matriz[i]==numero)
            |           indice = i
            FIN SI
        FIN SI
    FIN SI
FIN PARA

MENSAJE "la posición de " + numero + " es " + indice
```

FIN

No 2 - Given an array of integers, return an array of non-duplicate / unique elements.

Example

Test Case 1: [1,3,3,3,1,5]

Expected Output: [1,3,5]

Test Case 2: []

Expected Output: []

Test Case 3: [1,1,1,1]

Expected Output: [1]

INICIO

VARIABLE matriz1[] entero, matriz2[] entero, cantidad entero

MENSAJE "ingrese la cantidad de datos que contendrá la matriz"

RECIBIR (cantidad)

PARA i de 0 hasta (cantidad-1)

 matriz1[i] = aleatorio(10)

FIN PARA

ORDENAR (matriz1)

VARIABLE j entero

j = 0

PARA i de 0 hasta (cantidad-1)

 SI matriz1[i] != matriz1[i+1]

 matriz2[j++] = matriz1[i]

 FIN SI

FIN PARA

SI matriz2[cantidad-1] != matriz2[j]

 matriz2[j++] = matriz1[cantidad - 1]

FIN SI

MENSAJE "la matriz resultante es: " + matriz2

FIN

No 3 - Realizar un algoritmo que lea una cadena y determine si es un número válido.

- Ejemplos válidos: 34, 43.45, -42
- Ejemplos inválidos: 3a4, 43.4.5, 4-2-

INICIO

```
VARIABLE numero caracter
MENSAJE "ingrese un numero"
RECIBIR (numero)
EXPRESION REGULAR = ( ^\-[0-9]+\.[0-9]+$)
VARIABLE resultado carácter
SI (EXPRESION REGULAR (numero) == Verdadero)
    |
    |   Resultado == "Válido"
SI NO
    |
    |   Resultado == "Inválido"
FIN SI
MENSAJE "el número ingresado es " + Resultado
```

FIN

No4. Piramide de numeros

Realizar un algoritmo que genere un array aleatorio de 10 números de 1 a 100. Se sumarán todos los pares de números adyacentes para crear un nuevo array de 9 números. El proceso se repetirá creando arrays más pequeños hasta que solo quede un array de un número. Mostrar la pirámide generada

Ejemplo con un array inicial de 4 números

Array Inicial : [3, 5, 2, 1]
Resultado:

```
3      5      2      1
  8      7      3
    15     10
      25
```

INICIO

```
VARIABLE matriz[10,10] entero, m entero

PARA i de 0 hasta 9
|
|   matriz[1,i] = aleatorio(100)
|
FIN PARA

PARA i de 1 hasta 9
|
|   PARA j de 0 hasta 9
|   |
|   |   m = 0
|   |   |
|   |   |   PARA k de 0 hasta 9 en 2
|   |   |   |
|   |   |   |   matriz[i,m] = matriz[j,k] + matriz[j,k+1]
|   |   |   |   m++
|   |   |   |
|   |   |   FIN PARA
|   |   |
|   |   FIN PARA
|   |
|   FIN PARA
|
FIN PARA

PARA i de 0 hasta 9
|
|   PARA j de 0 hasta 9
|   |
|   |   MENSAJE matriz[i,j]
|   |
|   FIN PARA
|
FIN PARA
```

FIN