Práctico 6 Bis - Arreglos y Matrices

Arreglos o vectores

**Ejercicio 1.1**

**Ingresar 5 números en un arreglo de dimensión 5 y luego mostrarlos.**

#Ejercicio 1.1

import numpy as np

m=[[1],[2],[3],[4],[5]]

print(m)

**Ejercicio 1.2**

**Generar un arreglo de dimensión 20, llenarlo con valores al azar (entero aleatorio en un cierto rango) y mostrarlo luego de llenado.**

#Ejercicio 1.2

import random

import numpy as np

b=0

m=[[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]

for a in range (20):

b=random.randint (0,10)

m [a]=b

print(m)

**Ejercicio 1.3 (Arreglo de lógicos (Booleanos))**

**Diseñar un algoritmo que recorra las butacas de una sala de cine y determine cuántas butacas desocupadas hay en la sala. Suponga que inicialmente tiene un array (arreglo) con valores boolea- nos que si es verdadero (verdadero) implica que está ocupada y si es falso (falso) la butaca está desocupada. Tenga en cuenta que el arreglo (array) deberá ser creado e inicializado al principio del algoritmo**

#Ejercicio 1.3

import random

import numpy as np

numpy.random.choice(m,(False,True),size=(10,10)

for f in range (10):

for c in range (10):

if m [f,c]==False:

a=a+1

print("Cantidad de butacas desocupadas: ",a)

print(m)

**Ejercicio 1.4**

**Ingresar 10 números por teclado en dos arreglos alternadamente, 5 en cada uno, y luego mostrar los números ingresados todos juntos en el primer y en el segundo arreglo**

#Ejercicio 1.4

import random

import numpy as np

v1=np.random.randint (0,1,size=(5))

v2=np.random.randint (0,1,size=(5))

for c in range (5):

a=int(input("ingrese un número: "))

v1 [c]=a

b=int(input("ingrese un número: "))

v2 [c]=b

print(v1)

print(v2)

**Ejercicio 1.5**

**Generar un arreglo de dimensión 20, llenarlo con valores al azar, mostrar el contenido del arreglo luego de llenado, mostrar la suma de sus valores.**

#Ejercicio 1.5

import random

import numpy as np

b=0

d=0

m=[[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]

for a in range (20):

b=random.randint (0,10)

m [a]=b

print(m)

for c in range (20):

d=d+m [c]

print(d)

**Ejercicio 1.6**

**Generar un arreglo de dimensión 20, llenarlo con valores al azar, mostrar por separado el contenido sus posiciones pares y de las posiciones impares (trabajar sobre posiciones).**

#Ejercicio 1.6

import random

import numpy as np

b=0

m=[[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]

for a in range (20):

b=random.randint (0,10)

m [a]=b

print(m)

for c in range (20):

if c%2==0:

print ("Posición Par ", m [c])

else:

print ("Posición Impar ", m [c])

**Ejercicio 1.7**

**Generar un arreglo de dimensión 20, llenarlo con valores al azar, y decir cuántos valores pares contiene y cuantos impares (trabajar sobre contenido).**

#Ejercicio 1.7

import random

import numpy as np

b=0

m=[[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]

for a in range (20):

b=random.randint (0,10)

m [a]=b

print(m)

for c in range (20):

if m [c]%2==0:

print ("Número Par ", m [c])

else:

print ("Número Impar ", m [c])

**Ejercicio 1.8**

**Llenar un arreglo de dimensión 5 con número impares y luego mostrarlo en modo invertido.**

#Ejercicio 1.8

import numpy as np

b=0

m=[[1],[3],[5],[7],[9]]

n=4

for f in range (5):

print(m [n])

n=n-1

**Ejercicio 1.9**

**Ingresar 7 valores en un arreglo y posteriormente decir cuál es el mayor de ellos.**

#Ejercicio 1.9

import random

import numpy as np

v=np.random.randint (0,9,size=(7))

v.sort ()

print ("El número mayor es el ", v[6])

**Ejercicio 1.10**

**Buscar los dos mayores de una lista (arreglo) de *N* datos. *N* debe solicitarse por teclado.**

#Ejercicio 1.10

import numpy as np

v= []

a=int(input("Por favor, coloque la cantidad de números a ingresar "))

for i in range (a):

b=int(input("Por favor, ingrese los números "))

v.append(b)

v.sort ()

a=a-1

print ("El número mayor es el ", v[a])

**Ejercicio 1.11**

**Determinar el promedio de 10 notas ingresadas en un arreglo, indicando aprobado o des­aprobado. Siendo el aprobado un mayor al 60%.**

#Ejercicio 1.11

import numpy as np

v= []

a=0

for i in range (10):

b=101

while (b<0 or b>100):

b=int(input("Por favor, ingrese las notas "))

v.append(b)

for c in range (10):

a=a+v[c]

d=a/10

if d>= 60:

print ("Aprobado")

else:

print("Desaprobado")

**Ejercicio 1.12**

**Una escuela tiene un total de 3 aulas con la siguiente capacidad:**

|  |  |
| --- | --- |
| Identificador Aula | Cantidad de bancos del Aula |
| Azul | 40 |
| Verde | 35 |
| Amarillo | 30 |

**Sabiendo la cantidad de bancos de cada aula, el usuario deberá ingresar la cantidad de alumnos inscriptos para cursar tercer grado y el sistema deberá determinar qué aula es la indicada para la cantidad ingresada. La escuela ya sabe que la máxima capacidad de sus aulas es de 40 alumnos, por lo tanto, la cantidad de alumnos inscriptos que ingresa el usua­rio siempre será un número menor o igual a 40.**

**Listas necesarias para resolver el problema:**



**Ejercicio 1.13**

**Dada las siguientes notas almacenadas en un arreglo:**

**[33, 11, 20, 2, 15, 1, 12, 11, 8, 14, 10]**

**Elimine la nota más baja programáticamente sin usar la función (min) y escribala en pantalla. Luego programáticamente calcule el promedio de notas descontando la nota eliminada**

#Ejercicio 1.13

import numpy as np

v= [(33,11,20,2,15,1,12,11,8,14,10)]

v2=[len(v)-1]

a=0

v.sort ()

print (v[0])

i=1

for i in range (len(v)):

v2.append(v[i])

for c in range (len(v2)):

a=a+v2[c]

d=a/(len(v2))

**Ejercicio 1.14**

**Pida al usuario un número *n* por teclado, y cree un arreglo de tamaño *n* que contenga números primos partiendo desde 2 (un *número primo* es un número natural mayor que 1 que tiene únicamente dos divisores positivos distintos: él mismo y el 1)**

#Ejercicio 1.14

import numpy as np

v= [(33,11,20,2,15,1,12,11,8,14,10)]

v2=[len(v)-1]

a=0

v.sort ()

print (v[0])

i=1

for i in range (len(v)):

v2.append(v[i])

for c in range (len(v2)):

a=a+v2[c]

d=a/(len(v2))

**Ejercicio 2.1**

**Ingresar 12 valores en un arreglo (matriz) de 4x3 y mostrarlo en pantalla**

#Ejercicio 2.1

import numpy as np

m=np.random.randint (0,1,size=(4,3))

for f in range (4):

for c in range (3):

a=int(input("ingrese un número: "))

m [f,c]=a

print(m)

**Ejercicio 2.2**

**Ingresar 20 valores al azar en un arreglo de 5x4 y mostrar las filas por un lado, y luego las columnas por otro.**

#Ejercicio 2.2

import numpy as np

m=np.random.randint (0,1,size=(5,4))

for f in range (5):

for c in range (4):

a=int(input("ingrese un número: "))

m [f,c]=a

print(m)

for f in range (5):

for c in range (4):

print(m[f,c])

**Ejercicio 2.3**

**Ingresar 12 valores en un arreglo de 4x3 y mostrar las sumas de las filas y luego la suma de las columnas.**

#Ejercicio 2.3

import numpy as np

m=np.random.randint (0,1,size=(4,3))

for f in range (4):

for c in range (3):

a=int(input("ingrese un número: "))

m [f,c]=a

print(m)

for f in range (4):

for c in range (3):

print(m[f,c])

print ("La suma de la fila 1 es: "; m[(0,0)]+m[(0,1)]+m[(0,2)])

print ("La suma de la fila 2 es: "; m[1,0]+m[1,1]+m[1,2])

print ("La suma de la fila 3 es: "; m[2,0]+m[2,1]+m[2,2])

print ("La suma de la fila 4 es: "; m[3,0]+m[3,1]+m[3,2])

print ("La suma de la columna 1 es: ";m[0,0]+m[1,0]+m[2,0])

print ("La suma de la columna 1 es: ";m[0,1]+m[1,1]+m[2,1])

print ("La suma de la columna 1 es: ";m[0,2]+m[1,2]+m[2,2])

**Ejercicio 2.4**

**Ingresar 12 valores en un arreglo de 4x3 y cargar la suma de las filas en un arreglo y luego la suma de las columnas en otro arreglo. Luego recorrer y mostrar ambos arreglos.**

#Ejercicio 2.4

import random

import numpy as np

a=np.random.randint (0,9,size=(4,3))

b=np.random.randint (0,9,size=(4))

c=np.random.randint (0,9,size=(3))

b [(0)]=a [(0,0)] + a [(0,1)]+ a[(0,2)]

b [(1)]=a [(1,0)] + a [(1,1)]+ a[(1,2)]

b [(2)]=a [(2,0)] + a [(2,1)]+ a[(2,2)]

b [(3)]=a [(3,0)] + a [(3,1)]+ a[(3,2)]

c [(0)]=a [(0,0)] + a [(1,0)]+ a[(2,0)]

c [(1)]=a [(0,1)] + a [(1,1)]+ a[(2,1)]

c [(2)]=a [(0,2)] + a [(1,2)]+ a[(2,2)]

print (a)

print (b)

print (c)

**Ejercicio 2.5**

**Se ingresan los precios de 5 artículos y las cantidades vendidas por una empresa en sus 4 sucur­sales.**

**Informar:**

1. **Las cantidades totales de cada artículo.**
2. **La cantidad de artículos en la sucursal 2.**
3. **La cantidad del artículo 3 en la sucursal**
4. **La recaudación total de cada sucursal.**
5. **La recaudación total de la empresa.**
6. **La sucursal de mayor recaudación.**

**Ejercicio 2.6**

**Dada una matriz de 5x5 (la misma puede ser completada programáticamente o solicitando al usuario sus 20 números), imprimirla y luego modificar todos los números de sus diagonales por 1. Imprimir la matriz resultante.**

**Ejercicio 2.7**

**Cree una matriz de 3x4 con número enteros (al azar o de manera estática), y luego cree su tras­puesta\*. Imprimir ambas matrices.**

**\* Una matriz traspuesta es el resultado de reordenar la matriz original mediante el cambio de filas por columnas y las columnas por filas en una nueva matriz**

#Ejercicio 2.7

import random

import numpy as np

a=np.random.randint (0,9,size=(5,5))

b=np.random.randint (0,9,size=(5,5))

b [(0,0)]=a [(0,0)]

b [(1,0)]=a [(0,1)]

b [(2,0)]=a [(0,2)]

b [(3,0)]=a [(0,3)]

b [(4,0)]=a [(0,4)]

b [(0,1)]=a [(1,0)]

b [(1,1)]=a [(1,1)]

b [(2,1)]=a [(1,2)]

b [(3,1)]=a [(1,3)]

b [(4,1)]=a [(1,4)]

b [(0,2)]=a [(2,0)]

b [(1,2)]=a [(2,1)]

b [(2,2)]=a [(2,2)]

b [(3,2)]=a [(2,3)]

b [(4,2)]=a [(2,4)]

b [(0,3)]=a [(3,0)]

b [(1,3)]=a [(3,1)]

b [(2,3)]=a [(3,2)]

b [(3,3)]=a [(3,3)]

b [(4,3)]=a [(3,4)]

b [(0,4)]=a [(4,0)]

b [(1,4)]=a [(4,1)]

b [(2,4)]=a [(4,2)]

b [(3,4)]=a [(4,3)]

b [(4,4)]=a [(4,4)]

print (a)

print (b)

**Ejercicio 2.8**

**Dada una matriz de 5x5 con valores enteros al azar entre 1 y 8 0 y 9, generar dos arreglos, uno determinando si aparece el valor mostrado en la matriz, y otro con la cantidad de veces que aparece cada número en la matriz. Los índices de ambos arreglos deben coincidir para el número y su contador. Luego mostrar por pantalla cada número y la cantidad de aparicio­nes que tiene.**

**Dada una matriz de 5x5 con valores enteros al azar ingresados programáticamente entre 0 y 9, generar dos arreglos:**

**uno con valores lógicos (booleanos) que representan si el valor asociado al índice del arreglo aparece en la matriz con el contenido verdadero o falso en caso contrario.**

**otro arreglo con la cantidad de veces que aparece cada número en la matriz, donde el índice del arreglo debe coincidir con el número que aparece en la matriz y su contador.**

**Luego mostrar por pantalla cada número y la cantidad de apariciones que tiene y si perte­nece a alguna celda de la matriz.**

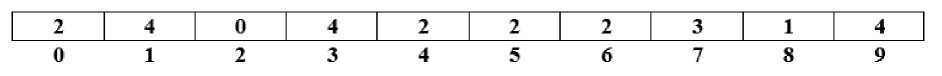
**Ejemplo: Matriz de 5x5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fila Columna | 0 | 1 | 2 | 3 | 4 |
| 0 | 0 | 7 | 8 | 5 | 3 |
| 1 | 7 | 0 | 1 | 4 | 5 |
| 2 | 1 | 9 | 3 | 7 | 9 |
| 3 | 6 | 1 | 3 | 9 | 4 |
| 4 | 9 | 1 | 3 | 7 | 6 |

Primer arreglo

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| True | True | False | True | True | True | True | True | True | True |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | S | 9 |

Segundo arreglo



**Ejercicio 2.9**

**Ingresar 12 valores en un arreglo a de 4x3 (al azar o de manera estática), y cargar la suma de las filas en un arreglo b y luego la suma de las columnas en otro arreglo c. Luego recorrer y mostrar ambos arreglos.**

#Ejercicio 2.9

import random

import numpy as np

a=np.random.randint (0,9,size=(4,3))

b=np.random.randint (0,9,size=(4))

c=np.random.randint (0,9,size=(3))

b [(0)]=a [(0,0)] + a [(0,1)]+ a[(0,2)]

b [(1)]=a [(1,0)] + a [(1,1)]+ a[(1,2)]

b [(2)]=a [(2,0)] + a [(2,1)]+ a[(2,2)]

b [(3)]=a [(3,0)] + a [(3,1)]+ a[(3,2)]

c [(0)]=a [(0,0)] + a [(1,0)]+ a[(2,0)]

c [(1)]=a [(0,1)] + a [(1,1)]+ a[(2,1)]

c [(2)]=a [(0,2)] + a [(1,2)]+ a[(2,2)]

print (a)

print (b)

print (c)