

Práctico 6 Arreglos

Arreglos implementados con listas

1. Ingresar 5 números en un arreglo de dimensión 5 y luego mostrarlos en pantalla.
2. Generar un arreglo de dimensión 20, llenarlo con valores al azar (*enteros aleatorios en un cierto rango*) y mostrarlo en pantalla al finalizar de llenarlo.
3. Arreglo de valores 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 booleanos que si es verdadero (*true*) implica que está ocupada y si es falso (*falso*) la butaca está desocupada. Tenga en cuenta que el arreglo deberá ser creado e inicializado al principio del algoritmo.
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.
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.
6. Generar un arreglo de dimensión 20, llenarlo con valores al azar, mostrar por separado el contenido de sus posiciones pares y de las posiciones impares (*trabajar sobre posiciones*).
7. Generar un arreglo de dimensión 20, llenarlo con valores al azar, y decir cuántos valores pares contiene y cuantos impares (*trabajar sobre contenido*).
8. Llenar un arreglo de dimensión 5 con número impares y luego mostrarlo en modo invertido.
9. Ingresar 7 valores en un arreglo y posteriormente determinar cuál es el mayor de ellos.
10. Buscar los dos mayores de una lista (*arreglo*) de N datos. Los N datos deben solicitarse por teclado.
11. Determinar el promedio de 10 notas ingresadas en un arreglo, indicando aprobado o desaprobado. Siendo el aprobado un promedio mayor al 60 %.

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:

Azul	Verde	Amarillo
0	1	2

40	35	30
0	1	2

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.
14. Pida al usuario un número n por teclado, y cree un arreglo de tamaño n que contenga números primos partiendo desde 2 hasta n (un número primo es un número natural mayor que 1 que tiene únicamente dos divisores positivos distintos: él mismo y el 1).
15. Ingresar 12 valores en un arreglo (matriz) de 4×3 y mostrarlo en pantalla.
16. Ingresar 20 valores al azar en un arreglo de 5×4 y mostrar las filas por un lado, y luego las columnas por otro.
17. Ingresar 12 valores en un arreglo de 4×3 y mostrar las suma de las filas y luego la suma de las columnas.

18. Ingresar 12 valores en un arreglo de 4×3 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.
19. Se ingresan los precios de 5 artículos y las cantidades vendidas por una empresa en sus 4 sucursales. Informar:
- a) Las cantidades totales de cada artículo.
 - b) La cantidad de artículos en la sucursal 2.
 - c) La cantidad del artículo 3 en la sucursal
 - d) La recaudación total de cada sucursal.
 - e) La recaudación total de la empresa.
 - f) La sucursal de mayor recaudación.
20. Dada una matriz de 5×5 (*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.
21. Cree una matriz de 3×4 con números enteros (*al azar o de manera estática*), y luego cree su traspuesta. Imprimir ambas matrices.
Nota: 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
22. Dada una matriz de 5×5 con valores enteros al azar ingresados programáticamente entre 0 y 9, generar dos arreglos:
- a) Uno con valores lógicos (*booleanos*) que representan si el valor asociado al índice del arreglo aparece en la matriz con el contenido entonces, será verdadero; falso en caso contrario.
 - b) 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 A continuación, se muestra un 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	8	9

Segundo arreglo

2	4	0	4	2	2	2	3	1	4
0	1	2	3	4	5	6	7	8	9

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

Matriz **a 4x3**

Fila/Columna	0	1	2
0	1	3	6
1	9	10	4
2	7	2	9
3	3	21	16

Arreglo **b** con la suma de las filas:

0	1	2	3
<u>b</u> [0] = 10	<u>b</u> [1] = 23	<u>b</u> [2] = 18	<u>b</u> [3] = 40

$$\underline{b}[0] = a[0][0] + a[0][1] + a[0][2] = 1 + 3 + 6 = 10$$

$$\underline{b}[1] = a[1][0] + a[1][1] + a[1][2] = 9 + 10 + 4 = 23$$

Arreglo **c** con la suma de las columnas:

0	1	2
<u>c</u> [0] = 20	<u>c</u> [1] = 36	<u>c</u> [2] = 35

$$c[0] = a[0][0] + a[1][0] + a[2][0] + a[3][0]$$

$$c[1] = a[0][1] + a[1][1] + a[2][1] + a[3][1]$$

Numpy Arrays

Nota: Si bien los arreglos se pueden implementar en particular con listas, cabe destacar que las listas son lentas cuando manejan muchos datos; una de las implementaciones más eficientes de arreglos, para Python, es la librería Numpy.

¿Cómo Resolver los ejercicios?

*Sin utilizar **ninguna** de las funciones sofisticadas que Numpy trae disponibles ni operaciones aritméticas sobre los arrays completos, deberá hacer uso de **bucles y el acceso individual a cada elemento a través de su posición dentro del array**, es decir, mediante el uso de índices.*

24. Pedirle 6 números enteros al usuario y guardarlos en una lista. Crear un array de una dimensión en base a dicha lista.
25. Supongamos tienes una lista con las alturas en cm de todos los miembros de tu familia, por ejemplo [181.5, 72., 34.7, 171.3, 160.1]. Crear un array y mostrar sus atributos, el tipo de datos, tanto del array como de sus elementos. Mostrar también el total de familiares cargados en el array.
26. Crear un array de 3 dimensiones, que tenga 3 matrices de 2 filas por 4 columnas. Llenelo con ceros.
27. Crear una matriz de (4, 6) con valores al azar que pertenecen al intervalo [0, 1).
28. Crear un vector con un total de 25 elementos equidistantes en el intervalo [1, 6].
29. Crear un vector con números enteros al azar entre 0 y 5. Luego reemplazar los 0 con el valor -1.
30. Dada una lista de 3 números enteros cualesquiera, y un vector con 3 números enteros cualesquiera. ¿Qué sucede si suma la lista a si misma, lista + lista, y si hace lo mismo con el vector? Haga la prueba y compare los resultados.
31. Crear una matriz de 3 x 3, con valores que van de 1 a 9.
32. Crear una matriz de 16 x 20 con números al azar, de algún tipo que le guste, distinto al tipo de dato por defecto *float64*.
33. Crear un array de 5 filas y 6 columnas, llenarlo con valores numéricos enteros, al azar entre 1 y 6. Luego, reemplazar todos los valores en la fila 5, por el valor 0.
34. Crear una función que realice la suma de dos arrays de dimensión 1 y devuelva el array resultante. Sin utilizar el operador + directamente, sino creando un algoritmo que hiciese la suma lugar a lugar. (**a pulmón*)
35. Crear una función que realice un producto entre dos arrays de dimensión 1 y devuelva el vector resultante. Sin utilizar el operador * directamente, sino creando un algoritmo que hiciese el producto lugar a lugar. (**a pulmón ; sin considerar ángulo alguno que hay en el producto vectorial o cruz de dos vectores, sino simplemente el producto de los valores lugar a lugar*)

36. Extender ahora el ejercicio anterior, a otra función que permita realizar la suma de elementos lugar a lugar, de dos arrays de dimensión 2.

37. Generar la siguiente matriz e imprimirla en pantalla:

```
1 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

38. Generar la siguiente matriz e imprimirla en pantalla:

```
1 1 1 1 1
0 0 0 0 0
1 1 1 1 1
0 0 0 0 0
1 1 1 1 1
```

39. Generar la siguiente matriz e imprimirla en pantalla:

```
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
```

Variado

40. Pedirle al usuario las notas de 3 exámenes, para 8 alumnos diferentes. Utilizar todo lo aprendido hasta ahora, como archivos de texto, creación de funciones propias, etc. Recuerde comentar el código y ser prolijo.

- Guardarlos en un archivo de texto llamado notas.txt
- Leer ese archivo de notas, y crear una matriz para guardar esas notas. Cada columna será una nota, cada fila representará un alumno.
- Calcular la nota promedio total, la nota promedio por alumno, y la nota promedio por examen.
- Determinar la mejor nota de cada alumno.
- Determinar la peor nota de cada alumno.
- Determinar cuál de los 3 exámenes fue donde hubo la mejor nota.
- Determinar cuál de los 3 exámenes fue donde hubo la peor nota.

41. Generar una matriz de 3 notas para 500 alumnos de primer año, de la carrera de arquitectura, y realizar las mismas cuentas que en el punto anterior, pero en esta matriz ahora.

REPASO Teórico

Ejemplo de cómo recorrer un arreglo bidimensional de 3 filas por 2 columnas:

Proceso LeerYMostrarMatriz

Definir matriz, fila, columna Como Entero; Dimension matriz[3,2];

// Leer datos

Para fila <- 0 Hasta 2 Hacer

 Para columna <- 0 Hasta 1 Hacer

 Escribir "Ingrese el valor de la fila:", fila+1, " columna: ", columna+1;

 Leer matriz[fila, columna];

 FinPara

FinPara

// Mostrar datos

Escribir "La matriz ingresada es:";

Para fila <- 0 Hasta 2 Hacer

 Para columna <- 0 Hasta 1 Hacer

 Escribir sin saltar matriz[fila, columna], " ";

 FinPara

 Escribir "";

FinPara

FinProceso