

### **Práctico N° 7**

**Tema:** Arreglos

**Duración:** 6 clases

**Ej. 1)** Completar el siguiente algoritmo para permitir al usuario ingresar 31 números reales en el arreglo denominado **tempMarzo** (que representa las temperaturas del mes de marzo), y luego visualizar por pantalla solo las temperaturas que superaron los 30°.

**Algoritmo** AltasTemperaturas

**Léxico**

CantMax = 31

TArregloDeReales = arreglo[1..CantMax] de R

tempMarzo ∈ TArregloDeReales

i ∈ Z

**Inicio**

...

**Fin**

**Ej. 2)** Desarrollar una acción que permita ingresar números enteros, dado el siguiente léxico:

N= 1000

Tnumeros= arreglo [1..N] de Z

Num ∈ Tnumeros

a) Cargar 10 valores únicamente. Perfil de la acción: Acción Carga (resultado: num ∈ Tnumeros, dato: nmax ∈ Z )

b) Cargar valores hasta que el usuario ingrese el valor 99999. Perfil de la acción:

Acción Carga (resultado: num ∈ Tnumeros). Si se alcanza a ingresar N-1 números en el arreglo, la acción debería colocar un 99999 en la posición 1000.-

c) Dado el siguiente léxico:

N= 1000

Telem= Z

Tarre= arreglo [1..N] de Telem

TData =< num ∈ Tarre, cant ∈ [0..N]>

Desarrollar una acción para cargar valores hasta que el usuario conteste N a la pregunta ¿Quiere cargar otro valor?.

Perfil de la acción: Acción Carga (resultado: listaNum ∈ TData)

Nota: analizar cómo será recorrido posteriormente el arreglo, para consultar sus datos, en cada uno de estos casos.

**Ej. 3)** Desarrollar un algoritmo que permita realizar opcionalmente las siguientes operaciones sobre un arreglo de caracteres:

- Mostrar por pantalla los caracteres desde la posición 1 hasta la mitad si es par o hasta la mitad menos uno si la cantidad de caracteres cargados en el arreglo es impar.
- Mostrar por pantalla los caracteres situados en posiciones impares
- Mostrar por pantalla los caracteres situados en posiciones cuyo valor sea un primo (1,2,3, 5, 7, 9, 13, etcétera)
- Mostrar por pantalla los caracteres desde la última posición hacia la primera (de atrás para adelante)

**Ej. 4)** Desarrollar un algoritmo que encuentre en un arreglo de números enteros, el mayor de todos los números almacenados y lo informe por pantalla.

**Ej. 5)** Desarrollar un algoritmo que permita contar la cantidad de vocales que hay en un arreglo car[1..50] de caracteres. Resolver según:

- Se conoce el número n de caracteres que tiene el arreglo siendo n menor o igual a 50.
- Los caracteres están en el arreglo formando una secuencia que finaliza con el carácter \* .

**Ej. 6)** Desarrollar un algoritmo que mediante una función calcule la suma de los números enteros que tiene un arreglo el cuál será pasado como parámetro, así como la cantidad de elementos que tiene cargados.

**Ej. 7)** Desarrollar una acción que reciba una variable de tipo arreglo de enteros, la cantidad de elementos a cargar en él y devuelva el arreglo cargado de números, generados en forma aleatoria.

Nota: para resolver este ejercicio se asume la existencia de una cierta función llamada random(n). El parámetro n debe ser un valor entero. Si se pasa un valor n como parámetro, la función devolverá un número entero pseudoaleatorio en el rango de 0 a n-1. Así por ejemplo random (10) dará por resultado un número entero comprendido entre 0 y 9. Para obtener números pseudoaleatorios en un intervalo determinado, por ejemplo en el rango de 3 a 8 inclusive, llame a la función de la siguiente forma: random (6) + 3

**Ej. 8)** En un arreglo se almacenan las calificaciones de N alumnos. Elabore un algoritmo que permita calcular e informar lo siguiente:

- El promedio general del grupo
- Número de alumnos aprobados y número de alumnos reprobados. Se considera aprobado a una nota igual o mayor a 6
- Porcentaje de alumnos aprobados y reprobados
- Número de alumnos cuya calificación fue mayor a 4.

**Ej. 9)** Desarrollar una función llamada capicúa: que reciba un arreglo y la cantidad de elementos que tiene cargados. La función debe devolver verdadero si los elementos equidistantes de los extremos del arreglo son iguales (es decir, se lee igual de izquierda a derecha y de derecha a izquierda). Ejemplo de casos en que debería devolver verdadero:

N	E	U	Q	U	E	N
---	---	---	---	---	---	---

A	C	U	R	R	U	C	A
---	---	---	---	---	---	---	---

**Ej. 10)** Desarrollar una función llamada espejo:

- que reciba un arreglo y la cantidad de elementos que tiene cargados, y devuelva un arreglo pero con los elementos en posiciones opuestas a la original tomando como referencia la primera posición.

Ejemplo: si el arreglo fuese

P	E	R	R	O
---	---	---	---	---

La aplicación de la función daría como resultado:

O	R	R	E	P
---	---	---	---	---

- Desarrolle una acción espejo, parametrizada, que reciba un arreglo **A** y devuelva en un arreglo **B**, el espejo del arreglo **A**.

**Ej. 11)** Desarrollar una acción que dado un arreglo A ya cargado, que contiene t números enteros positivos menores o iguales a 10, copie en otro arreglo B todos los valores del arreglo de A, ignorando los valores duplicados que se encuentran en A. Mostrar la cantidad de elementos copiados en el arreglo B y los elementos del mismo.

Ejemplo:

**A**

1	2	2	5	4	3	2	4	1
---	---	---	---	---	---	---	---	---

**B**

1	2	5	4	3				
---	---	---	---	---	--	--	--	--

En pantalla deberá aparecer: Cantidad total de elementos: 5 – Elementos: 1 2 5 4 3

Nota: defina los dos arreglos de la misma dimensión.

**Ej. 12)** Dado un arreglo A ya cargado con t números enteros positivos menores o iguales a 10, y ordenado de menor a mayor, desarrolle una acción que permita insertar un elemento, en el lugar que le corresponde para mantener el orden creciente.

**Ej. 13)** Desarrollar un algoritmo que permite cargar un arreglo de caracteres, y luego devuelva el arreglo eliminando los espacios en blanco, para esto debe desplazar los caracteres situados a la derecha del espacio en blanco, un lugar hacia la izquierda. Ejemplo sea A un arreglo de caracteres A=(c,a,s,a, ,a,m,a,r,i,l,l,a, ,c,a,s,a, ,a,z,u,l), debe quedar una vez eliminado los espacios en blanco: A=(c,a,s,a,a,m,a,r,i,l,l,a,c,a,s,a,a,z,u,l). Para resolver crear las acciones: cargarfrase(), eliminablancos() y mostrarfrase(), cada una adecuadamente parametrizada. Puede rehusar acciones creadas en ejercicios anteriores.

**Ej. 14)** Para administrar una lista de personas se requiere hacer diversas acciones que permitan ese trabajo. Para almacenar los nombres se utilizará un registro que contiene dos campos, uno con un arreglo donde se guardan los nombres y otro con la cantidad actual de nombres almacenados en el arreglo. La cantidad máxima que se podrá almacenar será de 1000 nombres. Para poder administrar la lista se deben desarrollar las siguientes acciones y/o funciones:

- Una función llamada Vacía que reciba como parámetro el registro (con el arreglo de nombres y la cantidad de

nombres cargados), y devuelva verdadero si la lista está vacía y sino debe devolver falso.

- b) Una función llamada Llena que reciba como parámetro el registro (con el arreglo de nombres y la cantidad de nombres cargados), y devuelva verdadero si el arreglo está completamente lleno (es decir si ya tiene 1000 nombres cargados) y sino debe devolver falso.
- c) Una acción que permita insertar un nombre en el arreglo. Para ello se pasará como parámetros el registro (con el arreglo de nombres y la cantidad de nombres cargados), y el nuevo nombre a insertar. El nuevo nombre se inserta siempre al final de la lista. Después de ejecutada la acción, la cantidad debe quedar incrementada en una unidad.
- d) Una acción que permita suprimir al primer nombre de la lista, no importa cuál es. Simplemente suprime el primero cada vez que la acción es ejecutada. El parámetro que debe recibir es el registro (con el arreglo de nombres y la cantidad de nombres cargados). Después de ejecutada la acción, la cantidad debe quedar decrementada en una unidad.
- e) Una acción que permita mostrar todos los nombres de la lista. Los parámetros que debe recibir son: el arreglo y la cantidad de nombres que tiene cargados.

**Ej. 15)** En un arreglo unidimensional se han almacenado personas de una comunidad. Cada persona es representada por un registro de dos campos, uno el apellido y el otro el DNI. Se solicita desarrollar una acción que reciba el arreglo, la cantidad de elementos cargados en él y un número de DNI, lo que debe hacer la acción es informar cuál es el nombre de la persona que tiene ese DNI y si no lo encuentra informar que no existe un persona con ese DNI. Se debe realizar dos versiones:

- suponga que los elementos en el arreglo están desordenados.
- Suponga que los elementos en el arreglo están ordenados por DNI.

**Ej. 16** En un arreglo unidimensional se han almacenado cosechas (una por cada año) desde el año 1900 hasta la actualidad. Cada cosecha posee el número de toneladas de soja, de maíz, de trigo, y de maní (por separado). Desarrollar un algoritmo que proporcione la siguiente información:

- Promedio anual de toneladas cosechadas a lo largo de la década del 40 y el 50 de cada tipo de semilla.
- ¿Cuántos años tuvieron una cosecha de trigo superior al promedio anual de la década del 40 y el 50?
- ¿Cuántos años tuvieron una cosecha de maíz inferior al promedio anual de la década del 40 y el 50?
- ¿Cuál fue el año en el que se produjo el mayor número de toneladas desde el 1900 hasta la actualidad, para cualquier tipo de cultivo?

## Tema: Arreglos bidimensionales (matrices)

**Ej. 17)** Desarrollar un algoritmo que permita ingresar datos a una matriz **C** de caracteres, con una dimensión de **n** filas y **m** columnas. Una vez que se ha cargado la misma se deberá mostrar los elementos correspondientes a las filas impares. Desarrollar una acción para la carga de la matriz y otra para mostrar lo pedido. Para la carga debe solicitar al usuario cuantas filas y cuántas columnas tendrá la matriz, no podrá superar las 9 filas, ni las 8 columnas.

A	B	C	D
E	2	B	F
D	C	A	H
F	E	F	I
I	L	K	J

Ejemplo: En pantalla deberá aparecer: A B C D, luego D C A H, y I L K J

**Ej. 18)** Dado un arreglo bidimensional **K** cargado con **n\*n** números enteros ( $n > 2$ ):

- a- Calcular la suma de los datos de la zona sombreada (mitad de filas por mitad de columnas mas uno).


b- Calcular la suma de la zona sombreada y mostrarla por pantalla:


c- Imprimir los datos de la zona sombreada en el orden indicado por la flecha:


**Ej. 19)** Una fábrica que cuenta con tres departamentos de producción lleva la contabilidad en una matriz llamada GASTOS de 3\*12 (3 filas y 12 columnas) dónde cada elemento representa el total de los gastos por mes y por departamento.

Se quiere calcular: el total de gastos por mes y el total de gastos anuales por departamento.

Mes Dpto.	1	2	3	4	5	6	7	8	9	10	11	12
Departamento 1	30	...	...	...	...	...	...	...	...	...	...	...
Departamento 2	28,5	...	...	...	...	...	...	...	...	...	...	...
Departamento 3	27	...	...	...	...	...	...	...	...	...	...	...

Es decir que se debería listar por pantalla:

Mes 1, Departamento 1, gasto \$ 30  
 Mes 1, Departamento 2, gasto \$28,5  
 Mes 1, Departamento 3, gasto \$27  
 Mes 2, Departamento 1, gasto ....  
 .....  
 Mes 12, Departamento 3, gasto .....  
  
 Departamento 1, gasto anual \$....  
 Departamento 2, gasto anual \$....  
 Departamento 3, gasto anual \$.....

**Ej.20)** Corazón = Cáscara: Dada una Matriz de [n x m] de enteros, con  $n \geq 3$  y  $m \geq 3$ , determinar si la sumatoria de las celdas límites (cáscara) es igual que la sumatoria las celdas del interior (todas menos las límites). Por ejemplo, para la siguiente matriz de (4x4)

	1	2	3	4
1	1	2	1	1
2	1	2	3	-1
3	-1	4	2	1
4	3	1	1	1

La respuesta es verdadero, ya que las celdas límites suman 11 al igual que las del interior.

**Ej. 21)** Desarrolle un algoritmo que sume dos matrices y guarde el resultado en una tercera. Las matrices para que puedan sumarse deben tener las mismas dimensiones. Ejemplo:  $m1 = ((12,13,14,10), (15,16,17,10), (18,19,20,10))$  y  $m2 = ((2,1,1,1), (2,1,1,1), (2,1,1,1))$   
 Resultado:  $m3 = (14,14,15,11), (17,17,18,11), (20,20,21,11)$  los elementos de  $m3[i,j] = m1[i,j] + m2[i,j]$

**Ej. 22)** Desarrolle un algoritmo que multiplique una matriz por un vector (arreglo unidimensional). Para que se pueda realizar este producto el vector debe tener tantos elementos como columnas tiene la matriz: Así el producto entre A de nxm y B de dimensión m da como resultado un vector C de n elementos. Ejemplo:  $A = ((12,13,14,10), (15,16,17,10), (18,19,20,10))$  y  $B = (2,1,1,1)$ , Resultado:  $C = (12x2+13x1+14x1+10x1, 15x2+16x1+17x1+10x1, 18x2+19x1+20x1+10x1) = (61,73,85)$ .

Nota: el producto de dos matrices solo es posible si la primer matriz tiene tantas columnas como filas tiene la segunda. Así el producto de dos matrices X de m x n, por una matriz Y de n x p, da como resultado una matriz Z de m x p. Cada elemento de Z será:  $Z_{jk} = X_{j1} Y_{1k} + X_{j2} Y_{2k} + \dots + X_{jn} Y_{nk}$

**Ej. 23) Suma de vecinos:** Dada una matriz de enteros, determinar si existe una posición (informando su posición) en la misma tal que el valor en dicha posición es igual a la suma de sus vecinos rectilíneos (superior, inferior, izquierdo y derecho). Cabe aclarar que las posiciones que se encuentran en algunos de los límites (o esquinas) de dicha matriz, sólo deben considerarse las posiciones existentes. En caso de existir más de una, simplemente aclare cuál retorna. Por ejemplo, para la siguiente matriz de (5x4)

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

Posibles resultados son

$[2,2]=4$ , ya que sus vecinos rectilíneos  $([1,2]=1 + [2,1]=1 + [3,2]=1 + [2,3]=1) = 4$

$[5,3]=6$ , ya que los 3 (está en el límite) vecinos rectilíneos  $([4,3]=1 + [5,2]=3 + [5,4]=2) = 6$

**Ej.24) Sopa de letras:** Dada una Matriz de  $[n \times m]$  caracteres y un Arreglo de  $l$  caracteres, con  $1 \leq n$  y  $1 \leq m$ , determinar si la palabra que forma el arreglo está presente en la matriz al menos una vez, de forma vertical u horizontal (cabe destacar que a diferencia de la sopa de letra original, no debe considerarse que la palabra pueda estar en forma diagonal o al revés). Se debe informar entre que posiciones está la palabra, si es que está. Por ejemplo, para la siguiente matriz de (5x4) y arreglo:

	1	2	3	4	5
1	i	a	t	q	u
2	g	q	u	e	r
3	t	u	h	i	a
4	v	e	u	e	i

  

q	u	e
1	2	3

Un posible resultado es desde  $[2,2]$  hasta  $[2,4]$

**Ej. 25)** Dada un arreglo de 365 elementos, llamado lluvia10  $[1..12, 1..31]$ , cuyos valores corresponden a las precipitaciones diarias en milímetros ocurridas a lo largo del año 2010 en una determinada ciudad; desarrollar un algoritmo que calcule:

- Cuál es la lluvia promedio por día a lo largo del año 2010 (resultado: un solo valor, se puede resolver con una función)
- Cuál es la lluvia promedio mensual en cada uno de los meses del año 2010. (resultado: un arreglo de 12 elementos)
- Cuál es la máxima precipitación y en que día y mes ocurrió.
- Cuál es la mínima precipitación y en que día y mes ocurrió.
- Dada una cantidad de días, determinar cuál es en el año el período de dicha cantidad de días con menos precipitaciones y cuál es el período con más precipitaciones.
- Dado un promedio  $p$ , determinar cuál es el período del año más largo que lo contiene con una diferencia menor a 10 mm.

**Ejemplo de programa en Pascal que emplea un arreglo bidimensional y los datos son incluidos directamente, definiendo el arreglo como una constante.**

### **Ejemplo**

Analizar que hace el siguiente programa:

```
Program Ejemplo1;
uses CRT;
Const
  matNum:ARRAY[1..3,1..3] of INTEGER=( (23,45,68),(34,99,12),(25,78,89) );
  (* esta es una forma cómoda para ingresar datos a un array en la etapa de prueba *)
Var
  i,j,valMax,posMaxFil,posMaxCol:integer;
Begin
  ClrScr;
  valMax:=matNum[1,1];
  posMaxFil:=1;
  posMaxCol:=1;
  for i:=1 to 3 do begin
    for j:=1 to 3 do begin
      if matNum[i,j] > valMax then begin
        valMax:=matNum[i,j];
        posMaxFil:=i;
        posMaxCol:=j;
      end;
    end;
  end;
  writeln('EL VALOR MAXIMO ES: ', valMax:3, ' EN LA POSICION:[' ,posMaxFil,',',posMaxCol,']');
  readkey
End.
```

### Plan mínimo de ejercicios a realizar en clases:

1era Clase: 1), 2.a), 2.c), 3.c), 3.d) y 5.b)

2da Clase: 6), 8.c), 9), 10)

6) y 8) Hacer en Laboratorio (llevar algoritmo resuelto)

3ra Clase: 11), 12) y 13) (Pedir a los alumnos que traigan resuelto para laboratorio el 18)

4ta Clase 14.c, 14e), 15.b) y 17)

18) Hacer en Laboratorio (llevar algoritmo resuelto)

5ta Clase 20), 21), 22)

6ta Clase 23) y 25)

Hacer en Pascal el ejercicio que dará la cátedra y entregarlo en la fecha que se fije al efecto.