

PRÁCTICA 7

Arreglos II: Matrices

Matriz[i,j] ▼

	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12

Aclaración: los ejercicios marcados con * se recomiendan realizar en forma obligatoria durante la semana correspondiente a la realización de la práctica, acorde a lo estipulado en el cronograma. Además, se recomienda consultar la solución realizada con los ayudantes durante la práctica y de ser posible, escribir el programa en Lazarus Pascal y probar su ejecución. El resto de los ejercicios es necesario realizarlos como parte del estudio y preparación para el parcial.

Objetivos de la práctica:

Se espera que el alumno logre:

- Resolver problemas de algorítmica clásica con vectores
 - Realizar operaciones clásicas sobre matrices como insertar y borrar filas y columnas.
 - Aplicar matrices (arreglos de dos dimensiones) en la resolución de problemas.
 - Utilizar matrices de registros y ejercitar su manipulación a través de las operaciones requeridas en problemas sencillos.
1. Realizar un módulo que reciba un vector V, un valor N y un número I, y que devuelva el mismo vector donde se insertó el valor N en la posición I.
 2. *Realizar un módulo que reciba un vector A y dos valores I, J. El módulo debe devolver el vector A del cual se suprimen las componentes que se encuentran entre las posiciones I y J (inclusive las componentes de dichas posiciones).
 3. *Realizar un módulo que reciba dos vectores A y B y un valor I, y que devuelva el vector A en el que se insertan las componentes del vector B entre las componentes I e I+1 del vector A. Se debe validar que el parámetro I esté en un rango permitido.
 4. Realizar un módulo que reciba un vector V ordenado y un valor N, y retorne verdadero si se encuentra dicho valor en el vector, o falso en caso contrario.
 5. *Dada una matriz de números enteros de 50 x 50, realizar un programa para:
 - a) Imprimir los elementos de las filas pares de la matriz.
 - b) Imprimir los elementos de las columnas impares de la matriz.
 - c) Imprimir los elementos de la diagonal principal.
 6. Un Astrónomo estudia las distancias entre las estrellas, y tiene almacenada la información de las 250 estrellas más cercanas a la Tierra. De cada estrella dispone el nombre, la superficie estimada, los nombres de sus gases principales (a lo sumo 10) y la distancia en kilómetros a la Tierra (valor entero). Se solicita realizar un programa que a partir de esta información informe:
 - a) El nombre de las 2 estrellas con mayor superficie.
 - b) La cantidad de estrellas con al menos 5 gases.
 - c) Informar el nombre de las estrellas en la cual la suma de los dígitos pares de la distancia a la tierra es mayor que la suma de los dígitos impares.
 7. *Dada una secuencia de 1000 *posts* correspondientes al mes pasado que fueron seleccionados para calcular algunas estadísticas. Sabiendo que cada *post* contiene información de la fecha y del texto (de un máximo 280 caracteres) en que fue publicado se pide calcular e informar:
 - a) La cantidad de hashtags (palabras que empiezan con el carácter '#') y usuarios (palabras que empiezan con carácter '@').
 - b) La cantidad de *posts* publicados cada día.
 - c) La cantidad promedio de palabras de un *post*.

8. *Realice un programa que lea la información de los empleados administrativos de la Facultad, sabiendo que a lo sumo existen 1200 empleados. De cada empleado se conoce: DNI, Apellido, Nombre, fecha de nacimiento (día, mes, año), área administrativa (existen 8 áreas nombradas de la A a H) y el número de oficina en la que trabaja (existen 20 oficinas que se numeran de 1 a 20). Además, se dispone de una estructura que almacena el bono de productividad (monto en pesos) de cada área y oficina. La lectura de la información finaliza cuando llega el empleado con DNI 0, el cual no se procesa. Luego de leer la información de los empleados, se pide:
- Informar el Apellido y Nombre de los empleados que trabajan en la oficina número 1.
 - Calcular e informar el total de empleados por área.
 - Informar el bono de productividad y DNI de los empleados del mes. Un empleado del mes tiene un DNI que cumple con que la suma de sus dígitos múltiplos de 3 es mayor a la suma de sus dígitos pares.
 - Informar el nombre de los empleados que cumplen años en un mes que se ingresa desde teclado.
9. *Una imagen digital es una estructura en forma de matriz donde cada elemento almacena un color (pixel). En una imagen RGB (Red, Green, Blue) este elemento de color es una mezcla de intensidades de los colores (canales) Rojo, Verde y Azul. Teniendo en cuenta la siguiente estructura:

type

```
Color = record
  r: integer;
  g: integer;
  b: integer;
end;

ImagenRGB = record
  ancho: integer;
  alto: integer;
  datos: array[1..1024, 1..1024] of Color;
end;
```

- Implemente módulos que permitan:
 - copiar el contenido de una imagen en otra.
 - rotar 90 grados una imagen.
 - “espejar” horizontalmente una imagen.
- Descargue de la Mediateca de IDEAS el archivo ProyectoImágenes.zip que contiene un proyecto Lazarus con la estructura anteriormente mencionada y módulos para cargar y guardar una imagen en el disco. Complete el proyecto escribiendo los módulos implementados en a) dentro del programa principal y utilice las funciones para cargar y guardar imágenes para comprobar visualmente los resultados. Los encabezados de estas son:

```
// carga en el parámetro img la imagen desde el archivo
// ejemplo: cargarImagen('casa.jpg', imagen)
procedure cargarImagen(archivo: string; var img: ImagenRGB);
// guarda la imagen del parámetro img en un archivo
// ejemplo: guardarImagen('casa-proc.jpg', imagen)
procedure guardarImagen(archivo: string; img: ImagenRGB);
```

10. *Construir un programa que implemente y use:
- un módulo que reciba una matriz y un valor N, y que retorne dicha matriz en la que se ha eliminado la N-ésima fila.
 - un módulo que reciba una matriz A, un vector B y un valor N, y que inserte el vector B como columna N-ésima de la matriz A.

Nota: realizar las validaciones correspondientes.

11. Siguiendo las tendencias en la edición de fotos, ahora deseamos poder crear filtros con nuestro editor de fotos en Pascal. Se desea crear un módulo “blurrearImagen(fuente, destino, intensidad)” el cual nos permita *blurrear* (“borronear”) una imagen fuente y guardarla en otra imagen destino (este ejercicio es continuación del ejercicio 9 de la parte B de la práctica 6)

Una imagen *blurreada* es aquella en la cual el color de cada píxel es el resultado del promedio de colores de los píxeles de su vecindad (Incluyendo el propio). La intensidad del efecto depende directamente de cuántos píxeles sean tomados en cuenta para este cálculo. Tenga en cuenta que la cantidad de píxeles que forman parte de la vecindad de un píxel puede disminuir si el mismo se encuentra en una esquina de la imagen.

Por ejemplo, si se quiere calcular el nuevo valor del píxel de la fila 3 columna 3 de la imagen original, con una intensidad de 2, sus valores serán:

1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5

← intensidad →

↑ intensidad ↓

$$r_{3,3} = \frac{r_{1,1} + r_{1,2} + r_{1,3} + r_{1,4} + r_{1,5} + \dots + r_{5,4} + r_{5,5}}{25}$$

$$g_{3,3} = \frac{g_{1,1} + g_{1,2} + g_{1,3} + g_{1,4} + g_{1,5} + \dots + g_{5,4} + g_{5,5}}{25}$$

$$b_{3,3} = \frac{b_{1,1} + b_{1,2} + b_{1,3} + b_{1,4} + b_{1,5} + \dots + b_{5,4} + b_{5,5}}{25}$$