**Fundamentos de Programación (17 de junio de 2016)**

**(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)**

***Ejercicio:*** construir un programa en ***C*** lo más modular posible (atendiendo a los criterios de modularidad) que se ajuste a la especificación dada y documentar el diseño preliminar con la definición de las nuevas tipologías de datos, el diagrama de módulos (estructura del programa) y las interfaces de los módulos (prototipos de funciones), y el diseño detallado con las definiciones de los respectivos sub-programas.

**Ejercicio:** Construir un programa para calcular y validar la inversa de una matriz cuadrada ***A***. El programa leerá inicialmente por teclado la dimensión ***n*** de la matriz ***A*** (máximo ***100x100***) y seguidamente los elementos de la misma. A continuación, calculará si ello es posible la inversa de dicha matriz siguiendo estrictamente el método descrito en la nota adjunta. Seguidamente se imprimirán en pantalla la matriz original y la matriz inversa (o un mensaje indicando que la matriz no es regular). Finalmente, el programa deberá comprobar la exactitud del resultado, multiplicando la matriz ***A*** por su inversa en cualquier orden para ver si coincide con la matriz identidad (***A.A-1=A-1.A=I***), considerando una precisión de ***10-10*** debido a los errores de redondeo que se producen en la aritmética de reales con doble precisión, presentando en pantalla el resultado de dicha comprobación. En caso de fallar la comprobación, indicar en qué elementos (nº de fila y nº de columna) se obtiene un resultado que difiere del correspondiente de la matriz identidad, y presentar dicho valor en pantalla con ***15*** cifras decimales.

***Nota***: para calcular la inversa de ***A***, hay que realizar una secuencia de operaciones elementales de fila hasta convertirla en una matriz identidad ***I***. Esas mismas operaciones aplicadas a la matriz identidad ***I*** producen la matriz inversa ***A-1***. Las operaciones elementales de fila son:

* **Iik**: intercambio 🡪 intercambiar las filas **i** y **k**.
* **Ei(x)**: escalado 🡪 multiplicar los elementos de la fila **i** por un número **x** no nulo.
* **Rik(x)**: reemplazo 🡪 sumar a una fila **i** otra fila **k** multiplicada por el número **x**.

La estrategia a seguir consiste en:

1. Bajar la escalera: fila a fila, desde la primera fila hasta la última fila, conseguir mediante operaciones elementales de fila unos en la diagonal principal y ceros debajo de la diagonal principal:
   * Si el correspondiente elemento de la diagonal principal de la fila actual (denominado pivote) es cero, habrá que intercambiar si ello es posible dicha fila por otra fila de las que están por debajo cuyo elemento en dicha posición no sea nulo. Si ello no fuera posible, entonces la matriz inicial no es regular (y por tanto no se puede invertir).

Si además se quiere reducir el error de redondeo, a menudo hay que realizar intercambios de filas aun cuando los elementos del pivote no sean cero. Estrategias de pivoteo (se deberá implementar la tercera estrategia):

* + - Pivoteo trivial: elegir la primera fila encontrada de las que están por debajo de la fila actual (incluida la actual) con un valor diferente de cero en la columna del pivote.
    - Pivoteo parcial (o de columna máxima): elegir la fila debajo de la actual (incluida la actual) con mayor valor absoluto en la columna del pivote.
    - Pivoteo parcial escalonado (o de escalado de columna): elegir la fila debajo de la actual (incluida la actual) con mayor valor relativo en la columna del pivote con respecto al valor máximo de los elementos de la fila. Si la fila actual es la ***i***, el procedimiento a seguir consistirá en los siguientes pasos:
      * Localizar el máximo valor absoluto de cada fila ***f***, para las filas ***i*** hasta ***n***:

Nótese que si todos los elementos de una fila son cero entonces la matriz no es invertible.

* + - * Dividir en valor absoluto cada elemento de la columna del pivote (***i***) por el máximo de cada fila:
      * La fila ***p*** escogida como pivote es la que cumple:

1. Subir la escalera: fila a fila, desde la última fila hasta la primera, conseguir ceros encima de la diagonal principal sin deshacer lo conseguido.

Ejemplos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** |  | **A** |  | **A** |  |
| **(A| I)** |  | **(A| I)** |  | **// E1(1)**  **R21(-1)**  **R31(-2)** |  |
| **I12** |  | **// E1(1)**  **R21(-1)**  **R31(-1)**  **R41(-1)** |  | **// E2(1)**  **R32(-1)** |  |
| **// E1(1)**  **//R21(0)**  **R31(-2)** |  | **// E2(1)**  **R32(-1)**  **R42(-1)** |  | **A** no es regular: no es posible conseguir un uno en la diagonal principal de la tercera fila | |
| **//E2(1)**  **R32(1)** |  | **// E3(1)**  **R43(-1)** |  |  | |
| **E3(1/4)** |  | **// E4(1)**  **R34(-1)**  **R24(-1)**  **R14(-1)** |  |
| **R23(-2)**  **R13(1)** |  | **R23(-1)**  **R13(-1)** |  |
| **R12(-1)** |  | **R12(-1)** |  |
| **A-1** |  | **A-1** |  |