Ejercicios de funciones Con Matrices

Juan David Torres Avila

02/09/2024

- **mostrarMenu()**: Muestra el menú de opciones para las operaciones con matrices.
- **obtenerEntradaMatriz()**: Recibe los elementos de una matriz desde la entrada del usuario.
- **sumarMatrices()**: Suma dos matrices y almacena el resultado en otra matriz.
- **restarMatrices()**: Resta una matriz de otra y almacena el resultado en una matriz.
- **multiplicacionEscalar()**: Multiplica cada elemento de una matriz por un escalar.
- **multiplicacionMatrices()**: Realiza la multiplicación de dos matrices y guarda el resultado.
- **transponerMatriz()**: Transpone una matriz, intercambiando filas por columnas.
- **invertirMatriz2x2()**: Calcula la inversa de una matriz 2x2, si es posible.
- **determinanteMatriz2x2()**: Calcula el determinante de una matriz 2x2.
- **imprimirMatriz()**: Imprime una matriz en formato de tabla.

```cpp
#include <iostream>
#include <limits>

using namespace std;

const int MAX_FILAS = 4;
const int MAX_COLUMNAS = 4;

// Prototipos de funciones
void mostrarMenu();
```

```
11 void obtenerEntradaMatriz(int matriz[][MAX_COLUMNAS],
       int filas, int columnas);
12 void sumarMatrices(int A[][MAX_COLUMNAS], int B[][
       MAX_COLUMNAS], int resultado[][MAX_COLUMNAS], int
       filas, int columnas);
13 void restarMatrices(int A[][MAX_COLUMNAS], int B[][
       MAX_COLUMNAS], int resultado[][MAX_COLUMNAS], int
       filas, int columnas);
14 void multiplicacionEscalar(int A[][MAX_COLUMNAS], int
       filas, int columnas, int escalar);
15 void multiplicacionMatrices(int A[][MAX_COLUMNAS], int
        B[][MAX_COLUMNAS], int resultado[][MAX_COLUMNAS],
       int filasA, int columnasA, int columnasB);
16 void transponerMatriz(int A[][MAX_COLUMNAS], int
       resultado[][MAX_COLUMNAS], int filas, int columnas)
       ;
17 void invertirMatriz2x2(int A[][MAX_COLUMNAS], int
       resultado[][MAX_COLUMNAS]);
18 int determinanteMatriz2x2(int A[][MAX_COLUMNAS]);
19 void imprimirMatriz(int matriz[][MAX_COLUMNAS], int
       filas, int columnas);
20
21 void mostrarMenu() {
22     cout << "\nOperaciones con matrices:" << endl;
23     cout << "1. Sumar matrices" << endl;
24     cout << "2. Restar matrices" << endl;
25     cout << "3. Multiplicar por escalar" << endl;
26     cout << "4. Multiplicar matrices" << endl;
27     cout << "5. Transponer matriz" << endl;
28     cout << "6. Invertir matriz (2x2)" << endl;
29     cout << "7. Determinante (2x2)" << endl;
30     cout << "8. Salir" << endl;
31     cout << "Ingrese su opci n: ";
32 }
33
34 void obtenerEntradaMatriz(int matriz[][MAX_COLUMNAS],
       int filas, int columnas) {
35     cout << "Ingrese los elementos de la matriz:" <<
           endl;
36     for (int i = 0; i < filas; i++) {
37         for (int j = 0; j < columnas; j++) {
38             cout << "Ingrese elemento [" << i << "]["
                   << j << "]: ";
39             cin >> matriz[i][j];
40         }
41     }
```

```
42 }
43
44 void sumarMatrices(int A[][MAX_COLUMNAS], int B[][
       MAX_COLUMNAS], int resultado[][MAX_COLUMNAS], int
       filas, int columnas) {
45     for (int i = 0; i < filas; i++) {
46         for (int j = 0; j < columnas; j++) {
47             resultado[i][j] = A[i][j] + B[i][j];
48         }
49     }
50 }
51
52 void restarMatrices(int A[][MAX_COLUMNAS], int B[][
       MAX_COLUMNAS], int resultado[][MAX_COLUMNAS], int
       filas, int columnas) {
53     for (int i = 0; i < filas; i++) {
54         for (int j = 0; j < columnas; j++) {
55             resultado[i][j] = A[i][j] - B[i][j];
56         }
57     }
58 }
59
60 void multiplicacionEscalar(int A[][MAX_COLUMNAS], int
       filas, int columnas, int escalar) {
61     for (int i = 0; i < filas; i++) {
62         for (int j = 0; j < columnas; j++) {
63             A[i][j] *= escalar;
64         }
65     }
66 }
67
68 void multiplicacionMatrices(int A[][MAX_COLUMNAS], int
       B[][MAX_COLUMNAS], int resultado[][MAX_COLUMNAS],
       int filasA, int columnasA, int columnasB) {
69     for (int i = 0; i < filasA; i++) {
70         for (int j = 0; j < columnasB; j++) {
71             resultado[i][j] = 0;
72             for (int k = 0; k < columnasA; k++) {
73                 resultado[i][j] += A[i][k] * B[k][j];
74             }
75         }
76     }
77 }
78
79 void transponerMatriz(int A[][MAX_COLUMNAS], int
       resultado[][MAX_COLUMNAS], int filas, int columnas)
```

```
    {
80      for (int i = 0; i < filas; i++) {
81          for (int j = 0; j < columnas; j++) {
82              resultado[j][i] = A[i][j];
83          }
84      }
85  }
86
87  void invertirMatriz2x2(int A[][MAX_COLUMNAS], int
        resultado[][MAX_COLUMNAS]) {
88      int det = determinanteMatriz2x2(A);
89      if (det == 0) {
90          cout << "Error: La matriz es singular (
                determinante es 0). No existe inversa." <<
                endl;
91          return;
92      }
93      resultado[0][0] = A[1][1];
94      resultado[0][1] = -A[0][1];
95      resultado[1][0] = -A[1][0];
96      resultado[1][1] = A[0][0];
97      multiplicacionEscalar(resultado, 2, 2, 1 / det);
98  }
99
100 int determinanteMatriz2x2(int A[][MAX_COLUMNAS]) {
101     return (A[0][0] * A[1][1]) - (A[0][1] * A[1][0]);
102 }
103
104 void imprimirMatriz(int matriz[][MAX_COLUMNAS], int
        filas, int columnas) {
105     cout << "Matriz resultante:" << endl;
106     for (int i = 0; i < filas; i++) {
107         for (int j = 0; j < columnas; j++) {
108             cout << matriz[i][j] << " ";
109         }
110         cout << endl;
111     }
112 }
113
114 int main() {
115     int filasA, columnasA, filasB, columnasB, opcion,
            escalar;
116     int matrizA[MAX_FILAS][MAX_COLUMNAS], matrizB[
            MAX_FILAS][MAX_COLUMNAS], resultado[MAX_FILAS][
            MAX_COLUMNAS];
117
```

```
118     do {
119         mostrarMenu();
120         cin >> opcion;
121 
122         if (opcion >= 1 && opcion <= 8) {
123             switch (opcion) {
124                 case 1:  // Sumar matrices
125                     cout << "Ingrese el n mero de
                            filas para la matriz A (m ximo
                             4): ";
126                     cin >> filasA;
127                     cout << "Ingrese el n mero de
                            columnas para la matriz A (
                            m ximo 4): ";
128                     cin >> columnasA;
129 
130                     cout << "Ingrese el n mero de
                            filas para la matriz B (m ximo
                             4): ";
131                     cin >> filasB;
132                     cout << "Ingrese el n mero de
                            columnas para la matriz B (
                            m ximo 4): ";
133                     cin >> columnasB;
134 
135                     if (filasA != filasB || columnasA
                            != columnasB) {
136                         cout << "Error: Las matrices
                                deben tener las mismas
                                dimensiones para la suma."
                                << endl;
137                         break;
138                     }
139 
140                     cout << "Ingrese la matriz A:" <<
                            endl;
141                     obtenerEntradaMatriz(matrizA,
                            filasA, columnasA);
142                     cout << "Ingrese la matriz B:" <<
                            endl;
143                     obtenerEntradaMatriz(matrizB,
                            filasB, columnasB);
144                     sumarMatrices(matrizA, matrizB,
                            resultado, filasA, columnasA);
145                     imprimirMatriz(resultado, filasA,
                            columnasA);
```

```
146                     break;
147
148                 case 2:  // Restar matrices
149                     cout << "Ingrese el n mero de
                           filas para la matriz A (m ximo
                            4): ";
150                     cin >> filasA;
151                     cout << "Ingrese el n mero de
                           columnas para la matriz A (
                           m ximo 4): ";
152                     cin >> columnasA;
153
154                     cout << "Ingrese el n mero de
                           filas para la matriz B (m ximo
                            4): ";
155                     cin >> filasB;
156                     cout << "Ingrese el n mero de
                           columnas para la matriz B (
                           m ximo 4): ";
157                     cin >> columnasB;
158
159                     if (filasA != filasB || columnasA
                           != columnasB) {
160                         cout << "Error: Las matrices
                               deben tener las mismas
                               dimensiones para la resta."
                                << endl;
161                         break;
162                     }
163
164                     cout << "Ingrese la matriz A:" <<
                           endl;
165                     obtenerEntradaMatriz(matrizA,
                           filasA, columnasA);
166                     cout << "Ingrese la matriz B:" <<
                           endl;
167                     obtenerEntradaMatriz(matrizB,
                           filasB, columnasB);
168                     restarMatrices(matrizA, matrizB,
                           resultado, filasA, columnasA);
169                     imprimirMatriz(resultado, filasA,
                           columnasA);
170                     break;
171
172                 case 3:  // Multiplicar por escalar
173                     cout << "Ingrese el n mero de
```

```
                    filas para la matriz A (m ximo
                     4): ";
174                 cin >> filasA;
175                 cout << "Ingrese el n mero de
                        columnas para la matriz A (
                        m ximo 4): ";
176                 cin >> columnasA;
177                 cout << "Ingrese el valor escalar:
                     ";
178                 cin >> escalar;
179
180                 cout << "Ingrese la matriz A:" <<
                        endl;
181                 obtenerEntradaMatriz(matrizA,
                        filasA, columnasA);
182                 multiplicacionEscalar(matrizA,
                        filasA, columnasA, escalar);
183                 imprimirMatriz(matrizA, filasA,
                        columnasA);
184                 break;
185
186             case 4:  // Multiplicar matrices
187                 cout << "Ingrese el n mero de
                        filas para la matriz A (m ximo
                     4): ";
188                 cin >> filasA;
189                 cout << "Ingrese el n mero de
                        columnas para la matriz A (
                        m ximo 4): ";
190                 cin >> columnasA;
191
192                 cout << "Ingrese el n mero de
                        filas para la matriz B (m ximo
                     4): ";
193                 cin >> filasB;
194                 cout << "Ingrese el n mero de
                        columnas para la matriz B (
                        m ximo 4): ";
195                 cin >> columnasB;
196
197                 if (columnasA != filasB) {
198                     cout << "Error: El n mero de
                            columnas en la matriz A
                            debe ser igual al n mero
                            de filas en la matriz B
                            para la multiplicaci n."
```

```
                << endl;
199                 break;
200             }
201
202             cout << "Ingrese la matriz A:" <<
                    endl;
203             obtenerEntradaMatriz(matrizA,
                    filasA, columnasA);
204             cout << "Ingrese la matriz B:" <<
                    endl;
205             obtenerEntradaMatriz(matrizB,
                    filasB, columnasB);
206             multiplicacionMatrices(matrizA,
                    matrizB, resultado, filasA,
                    columnasA, columnasB);
207             imprimirMatriz(resultado, filasA,
                    columnasB);
208             break;
209
210         case 5:  // Transponer matriz
211             cout << "Ingrese el n mero de
                    filas para la matriz A (m ximo
                     4): ";
212             cin >> filasA;
213             cout << "Ingrese el n mero de
                    columnas para la matriz A (
                    m ximo 4): ";
214             cin >> columnasA;
215
216             cout << "Ingrese la matriz A:" <<
                    endl;
217             obtenerEntradaMatriz(matrizA,
                    filasA, columnasA);
218             transponerMatriz(matrizA,
                    resultado, filasA, columnasA);
219             imprimirMatriz(resultado,
                    columnasA, filasA);  // Imprime
                     la matriz transpuesta
220             break;
221
222         case 6:  // Invertir matriz (2x2)
223             cout << "Ingrese la matriz A (2x2)
                    :" << endl;
224             obtenerEntradaMatriz(matrizA, 2,
                    2);
225             invertirMatriz2x2(matrizA,
```

```
                        resultado);
226                     imprimirMatriz(resultado, 2, 2);
227                     break;
228
229                 case 7:  // Determinante (2x2)
230                     cout << "Ingrese la matriz A (2x2)
                            :" << endl;
231                     obtenerEntradaMatriz(matrizA, 2,
                            2);
232                     cout << "Determinante: " <<
                            determinanteMatriz2x2(matrizA)
                            << endl;
233                     break;
234
235                 case 8:
236                     cout << "Saliendo del programa."
                            << endl;
237                     break;
238             }
239         } else {
240             cout << "Opci n inv lida. Por favor
                    ingrese un n mero entre 1 y 8." <<
                    endl;
241             cin.ignore(numeric_limits<streamsize>::max
                    (), '\n'); // Limpia el buffer de
                    entrada
242         }
243     } while (opcion != 8);
244
245     return 0;
246 }
```