



Centro Universitario de los Altos

Programación Estructurada

Gael Gómez Rivera

12/03/24

```

Ingrese el tamaño de las matrices (entre 2 y 15): 14
¿Desea llenar las matrices con valores aleatorios? (S/N): S
Matriz A:
393 119 145 153 -36 245 294 -187 -169 -4 -141 -30 142 169
373 263 169 293 374 36 152 45 -174 254 87 104 368 235
107 366 -61 -182 -139 130 -43 64 375 -80 323 273 183 42
270 139 252 -110 -26 143 197 134 75 307 237 272 322 -126
279 236 65 -161 -139 -111 377 -43 106 255 -81 387 257 376
354 -59 354 99 269 233 -189 -70 -24 -10 286 170 -138 53
321 78 130 273 11 6 279 -102 314 153 -108 159 15 294
-159 -46 120 -86 190 -189 -89 325 -61 307 2 70 240 53
-166 180 -81 -38 -24 189 92 106 120 22 159 335 -168 200
-116 291 101 -50 335 340 111 191 282 -198 334 -49 215 72
185 182 70 81 -134 -153 57 92 268 -70 264 272 37 228
-30 105 -4 101 -33 37 124 -197 97 140 385 290 -80 305
236 232 -194 181 39 -116 303 31 26 -10 -26 238 383 -84
-200 336 245 -103 -133 14 305 -6 398 203 212 337 -173 36
Matriz B:
397 50 -20 -138 62 -35 -200 -109 -199 -142 312 14 331 -112
328 -142 229 87 -199 -193 -165 -100 -184 57 266 345 327 -117
-117 318 -65 206 -31 6 -56 -58 27 66 94 -29 -45 147
364 252 -136 197 -102 182 -163 153 -37 350 186 -23 -109 -118
280 398 21 -16 279 -80 -154 96 378 321 248 104 101 55
166 316 142 341 365 242 50 50 306 272 237 25 394 109
207 207 -112 62 100 352 295 126 -94 -98 194 -77 171 119
8 59 18 -152 24 -177 384 154 14 -74 212 283 253 194
41 70 60 266 102 350 -13 -168 257 118 283 51 -84 -99
15 -42 112 148 392 290 -146 253 338 -132 259 32 227 362
14 251 20 400 105 -98 211 373 103 -167 146 -40 288 56
225 -11 -55 163 235 -82 235 62 68 20 283 229 350 65
124 -92 -4 64 319 -167 265 181 -49 131 -32 -128 -190 177
123 110 -142 320 -19 398 254 -57 69 162 151 -60 96 -152
Selecione la operación a realizar:
1. Suma
2. Resta
3. Multiplicación
4. División (Matriz A / Matriz B)
Ingrese su opción: 1
Resultado de la suma:
790 169 125 15 26 210 94 -296 -368 -146 171 -16 473 57
701 121 398 380 175 -157 -13 -55 -358 311 353 449 695 118
-10 684 -126 24 -170 136 -99 6 402 -14 417 244 138 189
634 391 116 87 -128 325 34 287 38 657 423 249 213 -244
559 634 86 -177 140 -191 223 53 484 576 167 491 358 431
520 257 496 440 634 475 -139 -20 282 262 523 195 256 162
528 285 18 335 111 358 574 24 220 55 86 82 186 413
-151 13 138 -238 214 -366 295 479 -47 233 214 353 493 247
-125 250 -21 228 78 539 79 -62 377 140 442 386 -252 101
-101 249 213 98 727 630 -35 444 620 -330 593 -17 442 434
199 433 90 481 -29 -251 268 465 371 -237 410 232 325 284
195 94 -59 264 202 -45 359 -135 165 160 668 519 270 370
360 140 -198 245 358 -283 568 212 -23 121 -58 110 193 93
-77 446 103 217 -152 412 559 -63 467 365 363 277 -77 -116
-----
Process exited after 11.2 seconds with return value 0
Presione una tecla para continuar . . .

```

Código:

```
#include <iostream>

#include <ctime> // Para generar números aleatorios
#include <cstdlib> // Para generar números aleatorios


// Nombre: Gael Gómez Rivera
// Código: 423034247
// Centro Universitario de Los Altos


using namespace std;


const int Tmaximo = 15; // Tamaño máximo de la matriz


// Función para imprimir una matriz
void printMatrix(int matrix[Tmaximo][Tmaximo], int size) {
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            cout << matrix[i][j] << "\t";
        }
        cout << endl;
    }
}


// Función para llenar una matriz con valores aleatorios
void fillMatrixRandom(int matrix[Tmaximo][Tmaximo], int size, int seed) {
    srand(seed); // Inicializar la semilla para números aleatorios


    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            matrix[i][j] = rand() % 601 - 200; // Generar un número aleatorio entre -200
y 400
        }
    }
}
```

```
}
```

```
// Función para sumar dos matrices
```

```
void addMatrices(int matrixA[Tmaximo][Tmaximo], int matrixB[Tmaximo][Tmaximo], int  
result[Tmaximo][Tmaximo], int size) {
```

```
    for (int i = 0; i < size; ++i) {  
        for (int j = 0; j < size; ++j) {  
            result[i][j] = matrixA[i][j] + matrixB[i][j];
```

```
        }
```

```
    }
```

```
}
```

```
// Función para restar dos matrices
```

```
void subtractMatrices(int matrixA[Tmaximo][Tmaximo], int matrixB[Tmaximo][Tmaximo], int  
result[Tmaximo][Tmaximo], int size) {
```

```
    for (int i = 0; i < size; ++i) {  
        for (int j = 0; j < size; ++j) {  
            result[i][j] = matrixA[i][j] - matrixB[i][j];
```

```
        }
```

```
    }
```

```
}
```

```
// Función para multiplicar dos matrices
```

```
void multiplyMatrices(int matrixA[Tmaximo][Tmaximo], int matrixB[Tmaximo][Tmaximo], int  
result[Tmaximo][Tmaximo], int size) {
```

```
    for (int i = 0; i < size; ++i) {  
        for (int j = 0; j < size; ++j) {  
            result[i][j] = 0; // Inicializar el elemento de la matriz resultante en 0
```

```
            for (int k = 0; k < size; ++k) {  
                result[i][j] += matrixA[i][k] * matrixB[k][j];
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

// Función para dividir dos matrices

```
void divideMatrices(int numerator[Tmaximo][Tmaximo], int denominator[Tmaximo][Tmaximo],
int result[Tmaximo][Tmaximo], int size) {
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            if (denominator[i][j] != 0) {
                result[i][j] = numerator[i][j] / denominator[i][j];
            } else {
                cout << "Error: No se puede dividir por cero." << endl;
                return;
            }
        }
    }
}
```

```
int main() {
    int size;

    cout << "Ingrese el tamaño de las matrices (entre 2 y 15): ";
    cin >> size;

    if (size < 2 || size > 15) {
        cout << "Tamaño de matriz no válido. Debe estar entre 2 y 15." << endl;
        return 1;
    }
}
```

```
int matrixA[Tmaximo][Tmaximo];
int matrixB[Tmaximo][Tmaximo];
int result[Tmaximo][Tmaximo];
```

```
char option;

cout << "¿Desea llenar las matrices con valores aleatorios? (S/N): ";
cin >> option;
```

```
if (option == 'S' || option == 's') {
```

```
    fillMatrixRandom(matrixA, size, time(0)); // Usamos la hora actual como semilla
para matriz A
```

```
    fillMatrixRandom(matrixB, size, time(0) + 1); // Usamos la hora actual más 1
segundo como semilla para matriz B
```

```
    } else if (option == 'N' || option == 'n') {
```

```
        cout << "Ingrese los valores de la matriz A:" << endl;
```

```
        for (int i = 0; i < size; ++i) {
```

```
            for (int j = 0; j < size; ++j) {
```

```
                cout << "Ingrese el valor para la posición (" << i << ", " << j << "): ";
```

```
                cin >> matrixA[i][j];
```

```
            }
```

```
        }
```

```
        cout << "Ingrese los valores de la matriz B:" << endl;
```

```
        for (int i = 0; i < size; ++i) {
```

```
            for (int j = 0; j < size; ++j) {
```

```
                cout << "Ingrese el valor para la posición (" << i << ", " << j << "): ";
```

```
                cin >> matrixB[i][j];
```

```
            }
```

```
        }
```

```
    } else {
```

```
        cout << "Opción no válida." << endl;
```

```
        return 1;
```

```
    }
```

```
    cout << "Matriz A:" << endl;
```

```
    printMatrix(matrixA, size);
```

```
    cout << "Matriz B:" << endl;
```

```
    printMatrix(matrixB, size);
```

```
    cout << "Seleccione la operación a realizar:" << endl;
```

```
    cout << "1. Suma" << endl;
```

```
    cout << "2. Resta" << endl;
```

```
cout << "3. Multiplicación" << endl;
cout << "4. División (Matriz A / Matriz B)" << endl;
cout << "Ingrese su opción: ";
cin >> option;

switch(option) {
    case '1':
        addMatrices(matrixA, matrixB, result, size);
        cout << "Resultado de la suma:" << endl;
        printMatrix(result, size);
        break;
    case '2':
        subtractMatrices(matrixA, matrixB, result, size);
        cout << "Resultado de la resta:" << endl;
        printMatrix(result, size);
        break;
    case '3':
        multiplyMatrices(matrixA, matrixB, result, size);
        cout << "Resultado de la multiplicación:" << endl;
        printMatrix(result, size);
        break;
    case '4':
        divideMatrices(matrixA, matrixB, result, size);
        cout << "Resultado de la división:" << endl;
        printMatrix(result, size);
        break;
    default:
        cout << "Opción no válida." << endl;
        break;
}

return 0;
}
```