

```

/*
UNIVERSIDAD DE LAS FUERZAS ARMADAS "ESPE"
NOMBRES: ANA U, BRAYAN M, JORDY C.
GRUPO 4
MENU DE MATRICES (+(suma),-(resta),x(multiplicacion),A^x(potencia de una matriz))
*/

#include <stdio.h>

#define MAX 5

int main() {
    int n, opcion;

    int matrizA[MAX][MAX], matrizB[MAX][MAX], resultado[MAX][MAX];           //Definir
    variables locales de main

    int potencia;

    void leerMatriz(int matriz[MAX][MAX], int n, char nombre) {                //variables
    asignada parametros para optimizacion del programa

        printf("Ingrese los elementos de la matriz %c:\n", nombre);

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                printf("%c[%d][%d]: ", nombre, i, j);                        //Lectura de valores del
                usuario para filas y columnas

                scanf("%d", &matriz[i][j]);

            }
        }
    }

    void imprimirMatrizConNombre(int matriz[MAX][MAX], int n, char nombre) {
    //Funcion para imprimir con nombre las matrices (solo input)

```

```

printf("Matriz %c:\n", nombre);
for (int i = 0; i < n; i++) {
    printf("(");
    for (int j = 0; j < n; j++) {
        printf("%4d", matriz[i][j]);
    }
    printf(")\n");
}
}

```

```

void imprimirMatriz(int matriz[MAX][MAX], int n) {                //Funcion para
imprimir respuesta de operaciones (solo output)
    for (int i = 0; i < n; i++) {
        printf("(");
        for (int j = 0; j < n; j++) {
            printf("%4d", matriz[i][j]);
        }
        printf(")\n");
    }
}

```

```

void sumaMatrices(int a[MAX][MAX], int b[MAX][MAX], int res[MAX][MAX], int n) {    //
Suma de matrices
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            res[i][j] = a[i][j] + b[i][j];
}

```

```

void restaMatrices(int a[MAX][MAX], int b[MAX][MAX], int res[MAX][MAX], int n) {    //
Resta de matrices
    for (int i = 0; i < n; i++)

```

```

        for (int j = 0; j < n; j++)
            res[i][j] = a[i][j] - b[i][j];
    }

```

```

void multiplicarMatrices(int a[MAX][MAX], int b[MAX][MAX], int res[MAX][MAX], int n) { //
Multiplicacion de matrices

```

```

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            res[i][j] = 0;
            for (int k = 0; k < n; k++)
                res[i][j] += a[i][k] * b[k][j];
        }
}

```

```

void copiarMatriz(int origen[MAX][MAX], int destino[MAX][MAX], int n) { //Funcion
para almacenar matriz y asi multiplicar por si misma en void potencia

```

```

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            destino[i][j] = origen[i][j];
}

```

```

void potenciaMatriz(int matriz[MAX][MAX], int res[MAX][MAX], int n, int potencia) { //
Potencia de matriz

```

```

    int temp[MAX][MAX];
    copiarMatriz(matriz, res, n);
    for (int p = 1; p < potencia; p++) {
        copiarMatriz(res, temp, n);
        multiplicarMatrices(temp, matriz, res, n); //Llamamos a void copia
para la multiplicacion
    }
}

```

```

do {
    printf("=====GRUPO 4=====\\n");
    printf("=====MENU OPERADORES=====\\n");
    printf("====FUNDAMENTOS DE PROGRAMACION=====\\n");
    printf("\\nCalculadora de Matrices NxN (max 5x5)\\n");
    printf("Ingrese el tamaño N de las matrices (1 a 5): ");
    scanf("%d", &n);
    if (n < 1 || n > MAX) {
        printf("Error: El valor debe estar entre 1 y 5, intente de nuevo.\\n\\n\\n");
    }
} while (n < 1 || n > MAX);
header de programa // DO- WHILE para impresion de

```

```

do {
    printf("\\nMenu:\\n");
    printf("1. Sumar matrices\\n");
    printf("2. Restar matrices\\n");
    printf("3. Multiplicar matrices\\n");
    printf("4. Potencia de una matriz\\n");
    printf("0. Salir\\n");
    printf("Seleccione una opcion: ");
    scanf("%d", &opcion);
    para menu de opciones // DO- WHILE con switch interno

```

```

switch (opcion) {
    case 1: // Seleccion de 1
        leerMatriz(matrizA, n, 'A');
        leerMatriz(matrizB, n, 'B');
        imprimirMatrizConNombre(matrizA, n, 'A');
        imprimirMatrizConNombre(matrizB, n, 'B');
        sumaMatrices(matrizA, matrizB, resultado, n);
        printf("Matriz C: (A + B)\\n");

```

```

    imprimirMatriz(resultado, n);

    break;

case 2:                                     // Seleccion de 2

    leerMatriz(matrizA, n, 'A');

    leerMatriz(matrizB, n, 'B');

    imprimirMatrizConNombre(matrizA, n, 'A');

    imprimirMatrizConNombre(matrizB, n, 'B');

    restaMatrices(matrizA, matrizB, resultado, n);

    printf("Matriz C: (A - B)\n");

    imprimirMatriz(resultado, n);

    break;

case 3:                                     // Seleccion de 3

    leerMatriz(matrizA, n, 'A');

    leerMatriz(matrizB, n, 'B');

    imprimirMatrizConNombre(matrizA, n, 'A');

    imprimirMatrizConNombre(matrizB, n, 'B');

    multiplicarMatrices(matrizA, matrizB, resultado, n);

    printf("Matriz C :(A * B)\n");

    imprimirMatriz(resultado, n);

    break;

case 4:                                     // Seleccion de 4

    leerMatriz(matrizA, n, 'A');

    imprimirMatrizConNombre(matrizA, n, 'A');

    printf("Ingrese la potencia (entero positivo): ");

    scanf("%d", &potencia);

    if (potencia < 1) {

        printf("Potencia invalida.\n");

        break;

    }

    potenciaMatriz(matrizA, resultado, n, potencia);

    printf("Matriz C (A^%d):\n", potencia);

```

```
        imprimirMatriz(resultado, n);
        break;
case 0:                                     // Finalizar el programa
    printf("Saliendo del programa.\n");
    break;
default:
    printf("Opcion no valida.\n");
}
} while (opcion != 0);

return 0;                                  // Vuelve al menu
}
```