PSEUDO CODE

Nitbin S
221IT085

BEGIN

Function allocateMatrix
// Function to dynamically allocate matrix in heap
INPUT: No. of Rows and Columns
OUTPUT: Pointer to an array of Raw Pointers
matrix = Pointer to an array of Raw Pointers (Initialize ~~and allocate memory~~)

// Allocate memory for the array of raw pointers
matrix = Allocate memory for array of Integer pointers with size equal to raws
if matrix is NULL // If allocation Fails
Return NULL

// Loop to allocate memory for each raw pointer
For i ← 0 to raws do
matrix[i] = Allocate memory for array of integers with Size equal to columns
If matrix[i] is NULL // If allocation fails
Return NULL

Return matrix

Function multiplyMatrices

// Function to recursively multiply two matrices

INPUT: Double Pointer to Matrix A, B and Resultant Matrix
No. of Rows rA and Columns cA of Matrix A
No. of Rows rB and Columns cB of Matrix B

OUTPUT: Double Pointer to Resultant Matrix

Static Integer variables i=0, j=0, k=0

If i ≥ rA Then
    Return NULL
Else if i < rA Then
    If j < cB Then
        If k < cA Then
            R[i][j] += A[i][k] * B[k][j]
            k++
            Call multiplyMatrices(A, B, R, rA, cA, rB, cB)
        k=0
        j++
        Call multiplyMatrices(A, B, R, rA, cA, rB, cB)
    j=0
    i++
    Call multiplyMatrices(A, B, R, rA, cA, rB, cB)
Return R

Function printMatrix

// Function to print 2D array

INPUT : Double pointer to 2D Dynamic Array (matrix)
No. of Rows and Columns

OUTPUT : Void

For i ← 0 to rows

For j ← 0 to columns

Print matrix[i][j]

Print newline.

Function freeMatrix

// Function to free dynamically allocated memory in heap

INPUT : Double pointer to 2D Dynamic Array (matrix)
No. of Rows

OUTPUT : Void

For i ← 0 to rows

Free memory allocated to matrix[i]

Free memory allocated for matrix

Set matrix to NULL

Function fillRandomValues

// function to fill the 2D dynamic array with random values

INPUT: Double pointer to matrix
No. of rows & columns

OUTPUT: Void

For i ← 0 to rows
  For j ← 0 to cols
    value = Generate random integer between 1 & 100
    matrix[i][j] = value

Function Main

INPUT: None
OUTPUT: 0

A : Double Pointer to Matrix A
rA : No. of Rows of Matrix A
cA : No. of Columns of Matrix A
B : Double Pointer to Matrix B
rB : No. of Rows of Matrix B
cB : No. of Columns of Matrix B
R : Double Pointer to Resultant Matrix

Do

Print "Enter the dimensions of first matrix"
Read rA, cA
Print "Enter the dimensions of second matrix"
Read rB, cB
if (rA==0 or rB==0 or cA==0 or cB==0)
Print "Invalid Dimensions!"
Else If (rA<10 or rB<10 or cA<10 or cB<10)
Print "Minimum row and Column Dimension is 10"

While (rA<10 or rB<10 or cA<10 or cB<10)

If cA!=rB Then
Print "Matrix Multiplication is not possible!"
Return 0

A = allocateMatrix(rA, cA)

B = allocate Matrix (rB, cB)

R = allocate Matrix (rA, cB)

Seed random number generator with current time

fillRandomValues (A, rA, cA)

Seed random number generator with a random number

fillRandomValues (B, rB, cB)

Print "Matrix 1:"

printMatrix (A, rA, cB)

Print "Matrix 2:"

printMatrix (B, rB, cB)

R = multiplyMatrices (A, B, R, rA, cA, rB, cB)

Print "Result:"

printMatrix (R, rA, cB)

freeMatrix (A, rA)

freeMatrix (B, rB)

freeMatrix (R, rA)

Return 0

STOP