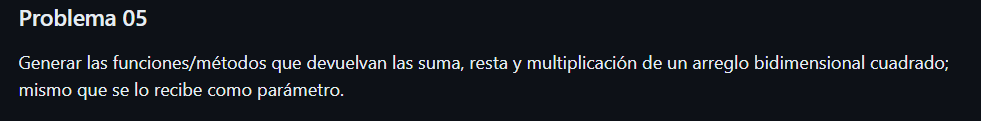
Ejercicio a resolver:



Análisis:

Como entrada necesitamos los limites de las matrices. Las operaciones se van mostrando en el flujo del programa. Para poder hacer los cálculos use procedimientos, en donde un procedimiento es solo la versión sobrecargada de el mismo procedimiento (sumarMatrices), y los presento a todos a través de una función que retorna valor en cadena (presentarMatrices), todos ellos son invocados desde el procedimiento principal o método main en java. Creo adicionalmente otros métodos/funciones para poder convertir las matrices de enteros a tipo decimal y asi poder ejercer división entre ellos.

Como salida, dependerá si fue la suma, resta o multiplicación de las matrices.

Pseudocódigo:

1. Algoritmo calculadoraMatrices
2. Definir tamanoMatriz Como Entero
3. Escribir "Ingrese la longitud de la matriz: "
4. Leer tamanoMatriz
5. Dimension mt1[tamanoMatriz,tamanoMatriz]
6. Dimension mt2[tamanoMatriz,tamanoMatriz]
7. Dimension matrizResultado[tamanoMatriz,tamanoMatriz]
8. Dimension mtdouble[tamanoMatriz,tamanoMatriz]
9. Dimension mtdouble2[tamanoMatriz,tamanoMatriz]
10. Dimension mtResultadodouble[tamanoMatriz,tamanoMatriz]
11. generarMatriz(mt1,tamanoMatriz)
12. generarMatriz(mt2,tamanoMatriz)
13. Escribir "La matriz 1 generada es:"
14. Escribir presentarMatrices(mt1,tamanoMatriz)
15. Escribir "La matriz 2 generada es:"
16. Escribir presentarMatrices(mt2,tamanoMatriz)
17. sumarMatrices(mt1,mt2,matrizResultado,tamanoMatriz)
18. Escribir "La suma de matrices es:"
19. Escribir presentarMatrices(matrizResultado,tamanoMatriz)
20. restarMatrices(mt1,mt2,matrizResultado,tamanoMatriz)
21. Escribir "La resta de matrices es:"
22. Escribir presentarMatrices(matrizResultado,tamanoMatriz)
23. multiplicarMatrices(mt1,mt2,matrizResultado,tamanoMatriz)
24. Escribir "La multiplicacion de matrices es:"
25. Escribir presentarMatrices(matrizResultado,tamanoMatriz)
26. FinAlgoritmo
27. SubProceso generarMatriz(mt1,tamanoMatriz)
28. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. mt1[i,j] <- Aleatorio(-9,10)
      2. FinPara
29. FinPara
30. FinSubProceso
31. Funcion cad = presentarMatrices(mt1,tamanoMatriz)
32. Definir cad Como Cadena
33. cad <- ""
34. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. cad <- cad + ConvertirATexto(mt1[i,j]) + " " + Tabulador
      2. FinPara
35. FinPara
36. FinFuncion
37. Funcion cad = presentarMatricesDouble(mtdouble,tamanoMatriz)
38. Definir cad Como Cadena
39. cad <- ""
40. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. cad <- cad + ConvertirATexto(mtdouble[i,j])+ " " + Tabulador
      2. FinPara
41. FinPara
42. FinFuncion
43. SubProceso sumarMatrices(mt1, mt2, matrizResultado,tamanoMatriz)
44. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. matrizResultado[i,j] <- mt1[i,j] + mt2[i,j]
      2. FinPara
45. FinPara
46. FinSubProceso
47. SubProceso restarMatrices(mt1, mt2, matrizResultado,tamanoMatriz)
48. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. matrizResultado[i,j] <- mt1[i,j] - mt2[i,j]
      2. FinPara
49. FinPara
50. FinSubProceso
51. SubProceso multiplicarMatrices(mt1,mt2, matrizResultado,tamanoMatriz)
52. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         + 1. matrizResultado[i,j] <- mt1[i,j] \* mt2[i,j]
      2. FinPara
53. FinPara
54. FinSubProceso
55. Funcion mtdouble = matrizInt\_a\_MatrizDouble(mt1, mtdouble, tamanoMatriz)
56. Para i<-0 Hasta tamanoMatriz-1 Hacer
    * 1. Para j<-0 Hasta tamanoMatriz-1 Hacer
         1. mtdouble[i,j] <- mt1[i,j] // ConvertirAReal
      2. FinPara
57. FinPara
58. FinSubProceso

Corrida de escritorio en PseInt:

