

Noah Meeker

821272563

CS 420

Assignment 3 Write-Up

For this assignment, we were task to write a program in Fortran that reads in two matrices and perform multiplication on them. The main requirements were the matrices were to be read from a file and two functions were to be used. All else was on the table as to how we would go about the final implementation. This current code reads in the file name of two text files. Each text file houses on matrix with the size of the matrix. After the multiplication, the final matrix is then written to a separate output file labeled "lab3_output.txt."

The read function reads in the size of the matrix on the first line. It then allocates a matrix based on the size parameters given. Once the matrix is read in, it returns it back to the main function.

The multiplication function takes in the two matrices after the program checks they are of the same size. The function allocates the final matrix based on the row size of matrix1 and the column size of matrix2. After the multiplication, the final matrix is returned to the main program.

There are checks to make sure each file is opened properly and the two matrices are of the correct sizes. The program prints the size of each input matrix and then print a completed statement to indicate the program finished. Below are screenshot examples.

TestFiles:

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more smalltest1.txt
3 3
1.0 2.0 3.0
4.0 5.0 6.0
7.0 8.0 9.0
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more smalltest2.txt
3 3
4.0 5.0 6.0
1.0 2.0 3.0
7.0 8.0 9.0
```

Run:

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ ./lab3
Enter Filename for first matrix
smalltest1.txt

Matrix: 3x3
Enter Filename for second matrix
smalltest2.txt

Matrix: 3x3
Multiplication Completed
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more lab3_output.txt
      21.00      24.00      27.00
      42.00      48.00      54.00
      63.00      72.00      81.00
```

TestFiles:

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more biggest1.txt
5 8
1.0 2.0 3.0 3.0 2.0 1.0 5.0 5.0
4.0 5.0 6.0 6.0 5.0 4.0 8.0 8.0
7.0 8.0 9.0 9.0 8.0 7.0 2.0 2.0
2.0 4.0 6.0 8.0 1.0 3.0 5.0 7.0
1.0 3.0 5.0 7.0 2.0 4.0 6.0 8.0
```

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more biggest2.txt
8 3
4.0 5.0 6.0
1.0 2.0 3.0
7.0 8.0 9.0
2.0 4.0 6.0
1.0 3.0 5.0
8.0 6.0 4.0
9.0 7.0 5.0
3.0 6.0 9.0
```

Run:

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ ./lab3
Enter Filename for first matrix
bigtest1.txt

Matrix: 5x8
Enter Filename for second matrix
bigtest2.txt

Matrix: 8x3
Multiplication Completed
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ more lab3_output.txt
      15.00      30.00      45.00
      24.00      48.00      72.00
       6.00      12.00      18.00
      21.00      42.00      63.00
      24.00      48.00      72.00
```

Proper matrix sizing for multiplication error handling:

```
(base) noah@pop-os:~/Documents/School_Work/SchoolProjects/CS420/lab3$ ./lab3
Enter Filename for first matrix
bigtest2.txt

Matrix: 8x3
Enter Filename for second matrix
bigtest1.txt

Matrix: 5x8
Matrix 1 Column Size does not match Matrix 2 Row Size
```

Current Code:

```
! PROGRAM LAB_3.F95
```

```
!
```

```
! Noah Meeker
```

```
! Assignment 3
```

```
PROGRAM LAB_3
```

```
IMPLICIT NONE
```

```
CHARACTER(20) :: filename
```

```
REAL, ALLOCATABLE :: matrix1(:, :), matrix2(:, :), matrix_final(:, :)
```

```
INTEGER, DIMENSION(2) :: size_arr
```

```
INTEGER :: idx, jdx, ios, count
```

```
! Grab first file name from user
```

```
PRINT *, "Enter Filename for first matrix"
```

```
READ *, filename
```

```
matrix1 = READ_MATRIX(filename)
```

```
! grab matrix size
```

```
PRINT *
```

```
WRITE(*, FMT=300) size(matrix1,1), size(matrix1,2)
```

```
! Grab second file name from user
```

```
PRINT *, "Enter Filename for second matrix"
```

```
READ *, filename
```

```
matrix2 = READ_MATRIX(filename)
```

```
PRINT *
```

```
WRITE(*, FMT=300) size(matrix2,1), size(matrix2,2)
```

```
! Check matrix sizes are correct
```

```
IF (size(matrix1,2) /= size(matrix2,1)) THEN
```

```
    PRINT *, "Matrix 1 Column Size does not match Matrix 2 Row Size"
```

```
    STOP
```

```
ENDIF
```

```
! Multiply the two matrices
```

```
matrix_final = MULT_MATRIX(matrix1, matrix2)
```

```
! Open output file
```

```
OPEN(UNIT=10, FILE="lab3_output.txt", ACTION="WRITE", STATUS="NEW",
```

```
IOSTAT=ios)
```

! Check output file opened successful

IF (ios /= 0) THEN

PRINT *, "Could not open output file"

STOP

ENDIF

! Print final matrix

DO idx = 1, size(matrix_final, 1)

DO jdx = 1, size(matrix_final, 2)

write(10, 100, ADVANCE="NO") matrix_final(idx, jdx)

END DO

write(10,*)

END DO

PRINT *, "Multiplication Completed"

100 FORMAT (f20.2)

300 FORMAT ("Matrix: ", i1, "x", i1)

CONTAINS

FUNCTION READ_MATRIX(filename) RESULT(matrix)

IMPLICIT NONE

CHARACTER(*), intent(in) :: filename

INTEGER :: ios, row, col

```
INTEGER, DIMENSION(2) :: matrix_size
```

```
REAL, ALLOCATABLE :: matrix(:,:)
```

```
! Open file
```

```
OPEN(UNIT=9, FILE=filename, ACTION="READ", STATUS="OLD", IOSTAT=ios)
```

```
! Check if file properly opened
```

```
IF(ios /= 0) THEN
```

```
    PRINT *, "Could not open file ", TRIM(filename), " Error code: ", ios
```

```
    STOP
```

```
ENDIF
```

```
! Grab matrix size
```

```
!
```

```
! matrix_size(1) = row
```

```
! matrix_size(2) = col
```

```
READ (9, *) matrix_size
```

```
ALLOCATE(matrix(matrix_size(1),matrix_size(2)))
```

```
! Read in matrix from file
```

```
DO row = 1, matrix_size(1)
```

```
    READ(9, *) (matrix(row, col), col = 1, matrix_size(2))
```

```
END DO
```

```
CLOSE(9)
```

```
END FUNCTION READ_MATRIX
```

```
! Function to multiply two matrices
```

```
FUNCTION MULT_MATRIX(matrix_a, matrix_b) RESULT(matrix_final)
```

```
    IMPLICIT NONE
```

```
    REAL, INTENT(IN) :: matrix_a(:, :), matrix_b(:, :)
```

```
    REAL, ALLOCATABLE :: matrix_final(:, :)
```

```
    INTEGER :: idx, jdx, kdx
```

```
    INTEGER :: m, n, p
```

```
    ! m = matrix1 row size
```

```
    ! n = matrix1 col size
```

```
    ! p = matrix2 row size
```

```
    m = size(matrix1, 1)
```

```
    n = size(matrix1, 2)
```

```
    p = size(matrix2, 2)
```

```
    ! Allocate final matrix size
```

```
    ALLOCATE(matrix_final(m,p))
```

```
    ! Multiply
```

```
    DO idx = 1, m
```

```
        DO jdx = 1, p
```

```
            DO kdx = 1, n
```

```
                matrix_final(idx, jdx) = matrix_a(idx, kdx) * matrix_b(kdx, jdx)
```

END DO

END DO

END DO

END FUNCTION MULT_MATRIX

END PROGRAM LAB_3