

## Hands #3 - Solving Problems with Matrices

### Problem Types:

- **Matrix Row and Column Manipulations**
- **Magic Squares**
- **Matrices for Text**

On the following pages, there are 9 sample problems. The dat files and solutions to these problems are provided in a separate folder.

The solutions are based on Java 5.0. I have used meaningful variable names to make the code more readable. Students in contest situations would certainly use shorter variable names and possibly more anonymous variables.

Problem	Key Concepts
pr31 - Matrix Fun	read numbers into a matrix; print out a matrix, manipulate a matrix
pr32 - Icicles	fill matrix one column at a time.
pr33 - Snowflake	fill matrix horizontally, vertically, and diagonally; change <code>int</code> to <code>char</code>
pr34 - Boxed-in 0	fill matrix horizontally and vertically from the outside in.
pr35 - Matrix Sums	find the sum of all the rows and columns
pr36 - Maze	search matrix - not by row/column; <code>toCharArray</code> ; while loops
pr37 - Add Matrices	traverse matrices; create new matrix
pr38 - Jumbled Matrix	<code>toCharArray</code> ; create new matrix; traverse alternating columns; print alternating rows
pr39 - LED Display	create a matrix of <code>char</code> ; insert chars in correct places in the matrix to form digits This problem is a definite 9 pointer.

## pr31 - Matrix Fun

<b>Problem:</b>	Write a program that will exchange the rows and columns of a square matrix
<b>Input:</b>	The first line of the data set is an integer that represents the number of data sets that follow. Each of the data sets will contain one line with one integer, which represents the number of rows and columns of the square matrix, followed by the rows of the matrix. The integers on each line are separated by a single space.
<b>data file:</b>	pr31
<b>Output:</b>	Output the matrix with row one and column one interchanged, row two and column two interchanged, etc. Both the original matrix and the "interchanged" matrix should be right justified in columns that are three characters wide. There should be at least one blank line between sets of output.
<b>Assumptions:</b>	All input will be integers in the range 1 .. 99.
<b>Sample Input:</b>	<pre>2 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 3 1 3 5 7 9 11 13 15 17</pre>
<b>Sample Output:</b>	<pre>1  5  9 13 2  6 10 14 3  7 11 15 4  8 12 16  1  7 13 3  9 15 5 11 17</pre>

## pr32 - Icicles

**Problem:** Write a program that will print the words of a sentence vertically in the order they appear in the sentence.

**Input:** The first line of the data set is an integer that represents the number of sentences that follow. Each of the following lines contains a sentence of less than 60 characters.

**data file:** pr32

**Output:** Output the words of each sentence vertically in the order they appear in the sentence. Print a blank column between the words. The words start at the top and go down. Print at least one blank line between data sets.

**Assumptions:** All sentences contain uppercase letters only.

**Sample Input:** 2  
MARY HAD A LITTLE LAMB  
THE FIRST WEEK IN FEBRUARY IS AWESOME

**Sample Output:**

```
M H A L L
A A   I A
R D   T M
Y     T B
      L
      E
```

```
T F W I F I A
H I E N E S W
E R E   B   E
  S K   R   S
    T     U   O
        A   M
        R   E
        Y
```

## pr33 - Snowflakes

**Problem:** Write a program that will print a snowflake shaped like an asterisk composed of the character given in the input.

**Input:** One line of positive integers with one space between each integer.

**data file:** pr33

**Output:** For each integer input, output a snowflake of that size.

**Assumptions:** Print at least one blank line between sets of output. Input will be in the range 4...9.

**Sample Input:** 6 4

**Sample Output:**

```
6      6      6
 6     6     6
   6    6    6
    6   6   6
     6  6  6
      666
66666666666666
      666
     6 6 6
    6 6 6
   6 6 6
  6 6 6
 6 6 6
```

```
4      4      4
 4     4     4
   4    4    4
    4   4   4
     444
44444444444444
     444
    4 4 4
   4 4 4
  4 4 4
 4 4 4
```

## pr34 - Boxed-in 0

**Problem:** Write a program to create a square matrix in which the outside square of the matrix is filled with the integer given and the inside squares decrease one with each level.

**Input:** One line of positive integers with one space between each integer.

**data file:** pr34

**Output:** For each integer input, output a square formatted like the ones below.

**Assumptions:** Print at least one blank line between sets of output. Input will be in the range 3...9.

**Sample Input:** 6 4

**Sample Output:**

```
66666666666666
65555555555556
65444444444456
65433333333456
65432222223456
6543211123456
6543210123456
6543211123456
6543222223456
6543333333456
6544444444456
6555555555556
66666666666666
```

```
4444444444
433333334
432222234
432111234
432101234
432111234
432222234
433333334
4444444444
```

## pr35 - Matrix Sums

<b>Problem:</b>	Write a program that will print the sums of all the rows and of all the columns of a matrix
<b>Input:</b>	The first line of the data set is an integer that represents the number of data sets that follow. Each data set will contain one line with two integers which represent the number of rows and columns of the matrix. The following lines will contain the matrix. The integers on each line are separated by a single space.
<b>data file:</b>	pr35
<b>Output:</b>	Output the original matrix with an extra row that contains the sum of each column and an extra column that contains the sum of each row. The element of the last row, column is the sum of all the rows. The elements of the matrix output should be right justified in columns that are four characters wide. There should be at least one blank line between sets of output.
<b>Assumptions:</b>	All input will be integers in the range 1 .. 99.
<b>Sample Input:</b>	<pre>2 4 5 1 2 3 4 5 5 6 7 8 9 9 10 11 12 13 13 14 15 16 17 5 6 1 3 5 7 9 10 7 9 11 13 15 14 13 15 17 19 21 2 4 5 6 7 8 5 3 6 9 2 5 8</pre>
<b>Sample Output:</b>	<pre> 1   2   3   4   5  15 5   6   7   8   9  35 9  10  11  12  13  55 13 14  15  16  17  75 28 32  36  40  44 180  1   3   5   7   9  10  35 7   9  11  13  15  14  69 13 15  17  19  21   2  87 4   5   6   7   8   5  35 3   6   9   2   5   8  33 28 38  48  48  58  39 259</pre>

## pr36 - Maze

<b>Problem:</b>	Write a program that will count the steps it will take to walk through a maze. The maze will contain # that represent walls and . to represent the path. Only horizontal and vertical steps may be taken. There will be only one choice of the direction to move from any position. There will be only one entry point which will be on the leftmost column of the maze. There will be only one exit point which will be in the rightmost column of the maze. There will be exactly one path through the maze.
<b>Input:</b>	The first line of the data set is an integer that represents the number of data sets that follow. Each data set will contain one line with two integers which represent the number of rows and columns of the matrix. The following lines will contain the maze.
<b>data file:</b>	pr36
<b>Output:</b>	Output the maze with a * that represents each step. After the maze, write PATH IS OF LENGTH followed by the number of steps there are in the path.
<b>Assumptions:</b>	None.

### Sample Input:

```
2
7 8
#####.
...####
###.###
###.#...
#.#.#.#
###...##
#####
6 9
#.#.###.#
#.#...###
##.##.#.
...##.##
#####
###.#####
```

### Sample Output:

```
#####.
****####
##*#####
###*#####
###*#####
#.#*#####
###*#####
#####
PATH IS OF LENGTH 14

#.#.###.#
#.#...###
##.##.#.
...##.##
#####
###.#####
PATH IS OF LENGTH 16
```

## pr37 - Adding Matrices

<b>Problem:</b>	Write a program that will add two matrices of the same dimensions.
<b>Input:</b>	The first line of the data set is an integer that represents the number of test cases that follow. The first line of each test case contains two integers that represent the number of rows and columns in each matrix. The remaining rows contain the matrices. There is exactly one space between elements on each row and no blank rows between matrices.
<b>data file:</b>	pr37
<b>Output:</b>	Output the sum of the two matrices of each test case. The elements of the matrix output should be right justified in columns that are four characters wide. There should be at least one blank line between sets of output.
<b>Assumptions:</b>	There are two matrices for each test case.
<b>Sample Input:</b>	<pre>2 3 4 1 2 3 4 5 6 7 8 9 10 11 12 12 11 10 9 8 7 6 5 4 3 2 1 4 6 6 5 4 9 8 7 9 6 3 2 5 8 7 4 1 2 5 8 6 5 4 9 8 7 10 11 12 13 14 15 3 6 9 9 6 3 8 5 2 2 5 8 7 4 1 1 4 7</pre>
<b>Sample Output:</b>	<pre>13 13 13 13 13 13 13 13 13 13 13 13  16 16 16 22 22 22 12 12 12 11 11 11 15 9 3 4 10 16 13 9 5 10 12 14</pre>



## pr38 - Jumbled Matrix

<b>Problem:</b>	Write a program that will jumble a matrix of letters as explained below.
<b>Input:</b>	The first line of the data set is an integer that represents the number of test cases that follow. The first line of each test case contains two integers that represent the number of rows and columns in each matrix. The remaining rows contain the matrix. There are no spaces between elements on each row.
<b>data file:</b>	pr38
<b>Output:</b>	Output the matrix that results when the odd number columns are placed first, followed by the even number columns. Then the odd numbered rows are placed first followed by the even numbered rows. Print at least one blank line between data sets.
<b>Assumptions:</b>	Row 0 is the first row and column 0 is the first column.
<b>Sample Input:</b>	<pre>2 6 5 ABCDE FGHIJ KLMNO PQRST UVWXY ZABCD 4 6 LETTER JACKET SCHOOL MASCOT</pre>
<b>Sample Output:</b>	<pre>ACEBD KMOLN UWYVX FHJGI PRTQS ZBDAC  LTEETR SHOCOL JCEAKT MSOACT</pre>

## pr39 - LED

**Problem:** Sally Sue runs the computer lab at CompSci University and wants to print a banner page for each program that a student prints from the lab. The banner page will contain the number of the computer that the student is using when he prints.

**Input file name:** pr39.dat

**Input:** The first line will contain the number of banner pages to be printed. Each of the following lines will contain two positive integers. The first integer will represent the `size` of the LED font, `size`  $\leq$  7. The second integer is the number of the computer that the student is printing from.

**Output:** For each printer number, the output is to be the computer number in a LED format. LED format is:

- \* Each digit is composed only of '-' and '|' characters as shown below
- \* Each digit occupies exactly `size + 2` columns, some of which may be all spaces
- \* Each digit occupies exactly `2*size + 3` rows
- \* There is exactly one column of spaces between each digit
- \* There is exactly at least one blank line between each set of output.

**Sample Input:**

```
3
2 12345
3 67890
4 281
```

**Sample Output:**

```

  --  --  --  --
  |  |  |  |  |  |
  |  |  |  |  |  |
  --  --  --  --
  |  |  |  |  |  |
  |  |  |  |  |  |
  --  --  --  --

  ---  ---  ---  ---  ---
  |  |  |  |  |  |  |
  |  |  |  |  |  |  |
  ---  ---  ---  ---
  |  |  |  |  |  |  |
  |  |  |  |  |  |  |
  ---  ---  ---  ---

  ---  ---
  |  |  |  |
  |  |  |  |
  ---  ---
  |  |  |  |
  |  |  |  |
  ---  ---
```