**Northeastern Illinois University**

**CS-200: Programming I**

**PLTL: 2D Arrays**

**Problem 1**

* Write a program that has the class name Problem1 and that has the main method. Leave

the main method empties for now.

* Write a method named transpose that takes one parameter, a 2-dimensional (2D) integer

array named a and returns a new 2D integer array.

* The method should create a new array a making the first column of a be the first row of

the new array, the second column of a be the second row of the new array, and so on. See

sample usage below.

* Create a printArray method that takes a 2D integer array as a parameter and prints out

the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the transpose method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] a = {{3, -2, 18},  {14, 0, 9}};  **int** [] [] a1 = transpose(a); | {{3, 14},  { -2, 0},  {18, 9}}; |
| **int** [] [] b = {{5, 8},  {6, 6},  {0, 4},  { -1, -2}};  **int** [] [] b1 = transpose(b); | {{5, 6, 0, -1},  {8, 6, 4, -2} }; |

**Problem 2**

* Write a program that has the class name Problem2 and that has the main method. Leave

the main method empties for now.

* Write a method named numberPatternArray that takes two parameters, an integer named

r and an integer named c. The method should return a 2-dimensional (2D) integer array.

* The method should create a new 2D array that has r rows and c columns. You can assume

that r and c will always be greater than 0. Determine the numerical pattern for each row

and assign the elements to the 2D array (see sample usage below). Return the array.

* Create a printArray method that takes a 2D integer array as a parameter and prints out

the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the numberPatternArray method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] a = numberPatternArray(3, 4); | {{0, 1, 2, 3},  {10, 11, 12, 13},  {20, 21, 22, 23}} |
| **int** [] [] b = numberPatternArray(9, 9); | {{0, 1, 2, 3, 4, 5, 6, 7, 8},  {10, 11, 12, 13, 14, 15, 16, 17, 18},  {20, 21, 22, 23, 24, 25, 26, 27, 28},  {30, 31, 32, 33, 34, 35, 36, 37, 38},  {40, 41, 42, 43, 44, 45, 46, 47, 48},  {50, 51, 52, 53, 54, 55, 56, 57, 58},  {60, 61, 62, 63, 64, 65, 66, 67, 68},  {70, 71, 72, 73, 74, 75, 76, 77, 78},  {80, 81, 82, 83, 84, 85, 86, 87, 88 }} |
| **int** [] [] c = numberPatternArray(5, 2); | {{0, 1},  {10, 11},  {20, 21},  {30, 31},  {40, 41}} |

**Problem 3**

* Write a program that has the class name Problem3 and that has the main method. Leave

the main method empties for now.

* Write a method named deepReverse that takes one parameter, a 2-dimensional (2D)

integer array named arr and returns a new 2D integer array.

* The method should create a new array a such that rows and columns are the reverse of

the array arr, such that first row of the array arr is the last row of the new array, second

row is the second last row of the new array and so on.

* Similarly, the first column in the array arr is the last column in the new array, second

column is the second last column in the new array and so on. See sample usage below.

* Create a printArray method that takes a 2D integer array as a parameter and prints out

the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the deepReverse method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] arr1 = {{1, 2, 4, 0},  {3, 4, 5, 6},  7, 8, 9, 12}};  **int** [] [] a1 = deepReverse(arr1); | {{12, 9, 8, 7},  {6, 5, 4, 3},  {0, 4, 2, 1}}; |
| **int** [] [] arr2 = {{2, 8},  {7, 20},  {9, 3},  {5, 12}};  **int** [] [] a2 = deepReverse(arr2); | {{12, 5},  {3, 9},  {20, 7},  {8, 2}}; |

**Problem 4**

* Write a program that has the class name Problem4 and that has the main method. Leave

the main method empties for now.

* Write a method named isPrime that takes one parameter, a 2-dimensional (2D) integer

array named arr and returns a new 2D boolean array.

* The method checks for every element in 2D array if it is a prime number or not. If it is a

prime, then element in boolean array at that index would be true else false.

* Create a printArray method that takes a 2D boolean array as a parameter and prints out the elements of each row on its own line separated by spaces.
* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the print2DArray method to print out the results of

calling the isPrime method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] a = {{4, 13, 10, 3, 9},  {14, 19, 43, 5},  {31, 17, 40, 11}};  **boolean** [] [] a1 = isPrime(a); | {{false, true, false, true, false},  {false, true, true, true},  {true, true, false, true} }; |
| **int** [] [] b = {{89, 7, 9},  {25, 39},  {133, 29, 41}};  **boolean** [] [] b1 = isPrime(b); | {{true, true, false},  {false, false},  {true, true, true} }; |

**Problem 5**

* Write a program that has the class name Problem5 and that has the main method. Leave

the main method empties for now.

* Write a method named binaryArr that takes one parameter, a 2-dimensional (2D)

boolean array named a and returns a 2D integer array.

* The method should create a new integer array that replaces every false from the booelan

array with a 0 and every true with a 1.

* Create a print2DArray method that takes a 2D integer array as a parameter and prints

out the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the transpose method!).

|  |  |
| --- | --- |
| Sample Method Usage | Output |
| **boolean** [] [] b1 = {{false, true, false, true, true},  {true, false, false, true, true},  {false, false, true, true, false}}; | 01011  10011  00110 |
| **boolean** [] [] b1 = {{true, false, true, false},  {false, false, true, false}}; | 1010  0010 |

**Problem 6**

* Write a program that has the class name Problem6 and that has the main method. Leave

the main method empties for now.

* Write a method named inSequence that takes a 2D integer array, and returns a boolean

value.

* The method should check if it the numbers are in sequence from 1 to n \* n. If they are,

then return true. You may assume that the array is already a perfect square.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code.

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] a1 = {{1, 2, 3, 4},  {5, 6, 7, 8},  {9, 10, 11, 12},  {13, 14, 15, 16}}; | **true** |
| **int** [] [] a1 = {{1, 3, 2, 4},  {3, 5, 8, 7},  {12, 11, 16, 8},  {12, 14, 15, 16}}; | **false** |