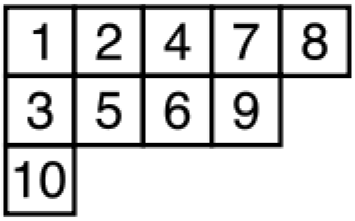
Project 6 (VNT) ***Updated June 21***
Summer 2012 CSCI 381/780 Adv. OOP in C++

Use your `Matrix` class (Project 4) to implement a VNT class.

Class `VNT` (stands for "Very Neat Table") utilizes a member m by n `Matrix` with increasing `int` entries in each row and column to represent a *Young's tableau* (example below). Constant `INT_MAX` is held in those `Matrix` positions for which there is not a box in the corresponding Young's diagram.



1	2	4	7	8
3	5	6	9	
10				

Thus, a `VNT` holds $r < mn$ `int`s of value less than `INT_MAX`.

The class should implement the following:

-- `VNT A(5,7);` --

1. `A[i][j]` returns the (i,j)th element of the underlying two-dimensional `Matrix`. So if the tableau is empty then `A[0][0]==INT_MAX` and if `A` is full then `A[m-1][n-1] <INT_MAX`.
2. Constructor `VNT(int m, int n)` which creates a tableau of at most m by n
3. `A.add(25)` will add `25` to a non-full `VNT`, returning `true` if the insertion is performed and `false` if the `Matrix` is full
4. `A.getMin()` which extracts the `A`'s smallest element and leaves `A` a `VNT`
5. `A.sort(int k[], int size)` sorts the $n \times n$ numbers in `k[]` using `VNT A` and does not call any sort routine as a subroutine
6. `A.find(int i)` which returns `true` if `i` is in `A` and `false` otherwise

<u>The Constraints</u>

a) function `void add(int)` should be $O(m+n)$
Hint: Start by inserting in bottom right corner. Compare index values and run swaps based on results. Repeat until terminating condition met.

b) function `int getMin()`should be $O(m+n)$

c) function `void sort(int [], int)` should be $O(n^3)$ for a n by n tableau

d) function `bool find(int)` should be $O(m+n)$
Hint: Start from top right or bottom left entry and run comparisons to determine entries for further comparison or termination.

These instructions supersede any and all instructions previously posted. If you have submitted a project which meets the requirements of a deprecated version of this assignment, there is no need to resubmit.