

HOMEWORK #1:

Planet of the Clams

For this assignment, you will submit a single C++ file called 'clams.cpp'.

Remember to put your name and section at the top of your program file.

Your program should expect all input to come from 'cin', and all your output should be to 'cout'.

NOTE: Start thinking about how your program will look like, but **do not** submit yet!!

On Monday (Aug 17) we will go over how to use the input files, how to test and how to submit your assignment.

Reminder: This homework is due Friday, August 31st.

Problem

Dr. Zoidberg is fishing for clams by walking along the bottom of the sea-floor. It is dark down there, so in order to know where to find the clams, a boat on the surface is sending him sonar data to his underwater smartphone. Each sonar data-pack represents the area ahead of Dr. Zoidberg as a grid, with the number of clams in each cell of the grid.

The sonar data is north oriented, and Dr. Zoidberg will walk across the bottom from south to north. Help Dr. Zoidberg choose which column of the sonar data grid to walk across in order to maximize the number of clams caught.

Input

The first line of the input gives the number of data grids **T**.

The first line of each test case contains the numbers **H** and **W**, the height and width of the sonar data grid. **H** rows of **W** data points follow, describing the number of clams detected by the sonar in each location of the sea floor.

Output

For each data grid, output one line containing "#x: y", where x is the grid number (starting from 0) and y is the westernmost (leftmost) column (starting from 0) in the fishing area where the most clams are present.

Implementation Requirements

Given that you do not know beforehand how large a sonar data grid is, your program should dynamically allocate a 2D Array after the height and width of a grid is read. Make sure to de-allocate the 2D Array after you find the column with the most clams and before your program

moves on to process the next grid.

Sample

Input	Output
3 2 2 5 2 0 9 2 6 4 1 2 0 0 6 0 7 5 9 8 3 3 3 1 0 0 0 1 0 0 2 1	#0: 1 #1: 3 #2: 1