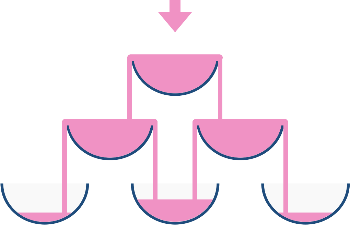
**17.We stack glasses in a pyramid, where the first row has 1 glass, the second row has 2 glasses, and so on until the 100th row.  Each glass holds one cup of champagne. Then, some champagne is poured into the first glass at the top.  When the topmost glass is full, any excess liquid poured will fall equally to the glass immediately to the left and right of it.  When those glasses become full, any excess champagne will fall equally to the left and right of those glasses, and so on.  (A glass at the bottom row has its excess champagne fall on the floor.) For example, after one cup of champagne is poured, the top most glass is full.  After two cups of champagne are poured, the two glasses on the second row are half full.  After three cups of champagne are poured, those two cups become full - there are 3 full glasses total now.  After four cups of champagne are poured, the third row has the middle glass half full, and the two outside glasses are a quarter full, as pictured below.**

****

**Now after pouring some non-negative integer cups of champagne, return how full the jth glass in the ith row is (both i and j are 0-indexed.)**

**Example 1:**

**Input: poured = 1, query\_row = 1, query\_glass = 1**

**Output: 0.00000**

**Explanation: We poured 1 cup of champange to the top glass of the tower (which is indexed as (0, 0)). There will be no excess liquid so all the glasses under the top glass will remain empty.**

**Example 2:**

**Input: poured = 2, query\_row = 1, query\_glass = 1**

**Output: 0.50000**

**Explanation: We poured 2 cups of champange to the top glass of the tower (which is indexed as (0, 0)). There is one cup of excess liquid. The glass indexed as (1, 0) and the glass indexed as (1, 1) will share the excess liquid equally, and each will get half cup of champange.**

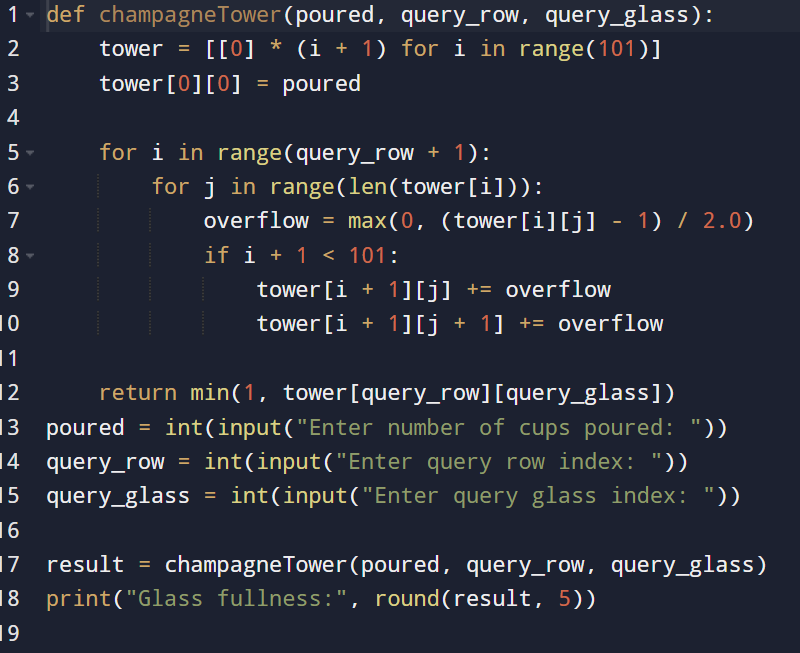
**Aim:**

To calculate how full a specific glass at position (query\_row, query\_glass) is, after pouring a given number of cups into the topmost glass of a champagne tower structured as a pyramid.

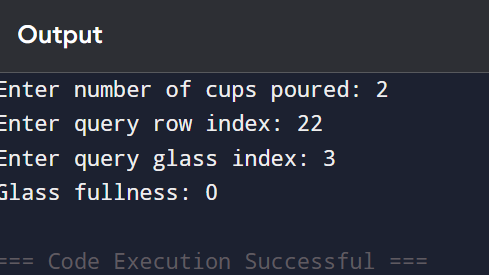
**Algorithm**:

1. Initialize a 2D array tower representing 100 rows, each with increasing number of glasses.
2. Set the top glass (0, 0) to poured cups.
3. For each glass, if it holds more than 1 cup, overflow is evenly distributed to the two glasses below it:
   * overflow = (tower[i][j] - 1) / 2
   * Add overflow to tower[i+1][j] and tower[i+1][j+1]
4. After simulating the pours, return the minimum of 1.0 and tower[query\_row][query\_glass] (since a glass can't be more than full).

**Code:**



**Input and output:**

****

**Result: given Champagne Tower – Track Fill Level of a Specific Glass in a Pyramid is successfully executed and output is verified.**

**Performance Analysis:**

* **Time Complexity: O(n2)*O*(*n*2), where n = \text{query\_row}**
* **Space Complexity: O(n2)*O*(*n*2), for the simulation array**