EX-1.17

Title :

Compute how full a specific glass is in the champagne tower after pouring cups of champagne.

Aim:

To design and implement a Python program to find how full the jth glass in the ith row is in a champagne tower after pouring a given number of cups.

Procedure:

1. Read input values: `poured` (number of cups poured), `query_row`, and `query_glass`.
2. Initialize a 2D array dp of size 101 x 101 (since max row is 100), with all entries 0.0 representing how full each glass is.
3. Pour the champagne into the top glass `dp = poured`.
4. For each row up to `query_row`:
    - For each glass in the row:
        - If `dp[i][j]` exceeds 1 cup, the excess `(dp[i][j] - 1)` will overflow equally to `dp[i+1][j]` (left glass) and `dp[i+1][j+1]` (right glass).
        - Set `dp[i][j]` to at most 1 (full).
5. After processing rows, the value in `dp[query_row][query_glass]` gives how full that glass is.
6. Print the fullness (float) rounded or formatted as needed.

Algorithm:

```
1. Start
2. Input poured, query_row, query_glass
3. Create dp 2D array with zeros
4. Set dp = poured
5. For i in 0 to query_row - 1:
       • For j in 0 to i:
             • excess = max(0, dp[i][j] - 1)
             • Distribute excess / 2 to dp[i+1][j] and dp[i+1][j+1]
             • Set dp[i][j] = min(1, dp[i][j])
6. Return dp[query_row][query_glass].
7. Stop
```

Input:

1 1 1

2 1 1

Output:

0.0

0.5

Program :

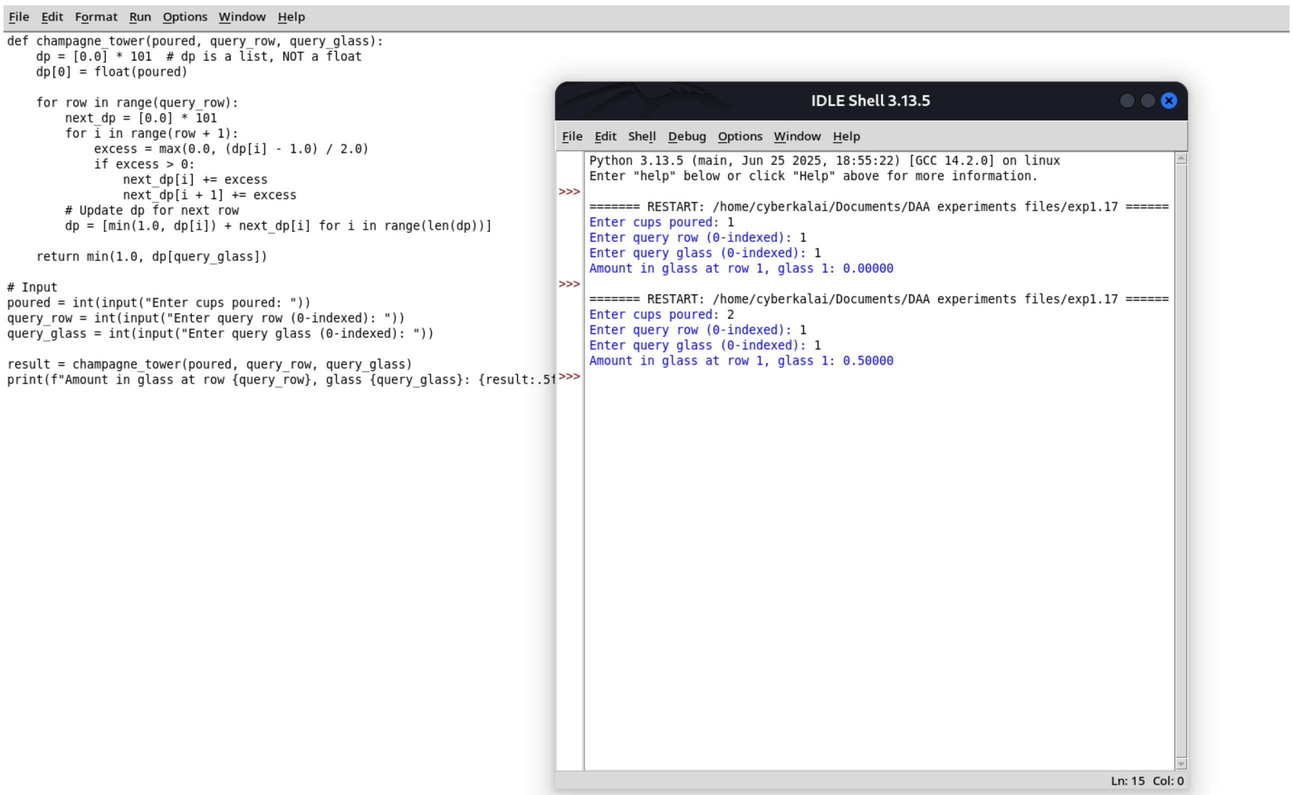
```
def champagneTower(poured, query_row, query_glass):
    dp = [[0.0] * 101 for _ in range(101)]
    dp = poured
    for i in range(query_row):
        for j in range(i + 1):
            excess = max(0.0, dp[i][j] - 1.0)
            if excess > 0:
                dp[i][j] = 1.0
                dp[i + 1][j] += excess / 2.0
                dp[i + 1][j + 1] += excess / 2.0
    return min(1, dp[query_row][query_glass])

poured, query_row, query_glass = map(int, input("Enter poured,
query_row, query_glass: ").split())

result = champagneTower(poured, query_row, query_glass)
print(f"{result:.5f}")
```

Performance Analysis:

Time Complexity: O(query_row^2)

Space Complexity: O(1)

program output:


File Edit Format Run Options Window Help
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Ln: 15 Col: 0

```
def champagne_tower(poured, query_row, query_glass):
    dp = [0.0] * 101  # dp is a list, NOT a float
    dp[0] = float(poured)

    for row in range(query_row):
        next_dp = [0.0] * 101
        for i in range(row + 1):
            excess = max(0.0, (dp[i] - 1.0) / 2.0)
            if excess > 0:
                next_dp[i] += excess
                next_dp[i + 1] += excess
        # Update dp for next row
        dp = [min(1.0, dp[i]) + next_dp[i] for i in range(len(dp))]

    return min(1.0, dp[query_glass])

# Input
poured = int(input("Enter cups poured: "))
query_row = int(input("Enter query row (0-indexed): "))
query_glass = int(input("Enter query glass (0-indexed): "))

result = champagne_tower(poured, query_row, query_glass)
print(f"Amount in glass at row {query_row}, glass {query_glass}: {result:.5f
```

```
Python 3.13.5 (main, Jun 25 2025, 18:55:22) [GCC 14.2.0] on linux
Enter "help" below or click "Help" above for more information.
>>>
======= RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.17 ======
Enter cups poured: 1
Enter query row (0-indexed): 1
Enter query glass (0-indexed): 1
Amount in glass at row 1, glass 1: 0.00000
>>>
======= RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.17 ======
Enter cups poured: 2
Enter query row (0-indexed): 1
Enter query glass (0-indexed): 1
Amount in glass at row 1, glass 1: 0.50000
>>>
```

Result :

Thus the given program Champagne Tower is executed and got output successfully.