Lesson:

2D Array Problems Set 2

PW SKILLS

Pre-Requisites

- Concept of Arrays and vectors
- Basic understanding of multidimensional arrays

List of Concepts Involved

- 2D Vectors Concept
- Pascal's Triangle Problem

Topic: Introduction to 2D Vectors

A 2D vector is a "vector of vector". Like 2D arrays, values can be declared and assigned to a 2D vector.

A normal vector is initialized as:

```
vector<datatype> vector_name;
```

Now in the case of a 2D vector, all we need to do is create a vector of datatype vector.

We simply replace "datatype" with "vector<datatype>":

```
vector<vector<datatype>> vector_name;
```

Initializing a 2-D vector with help of 1-D vector

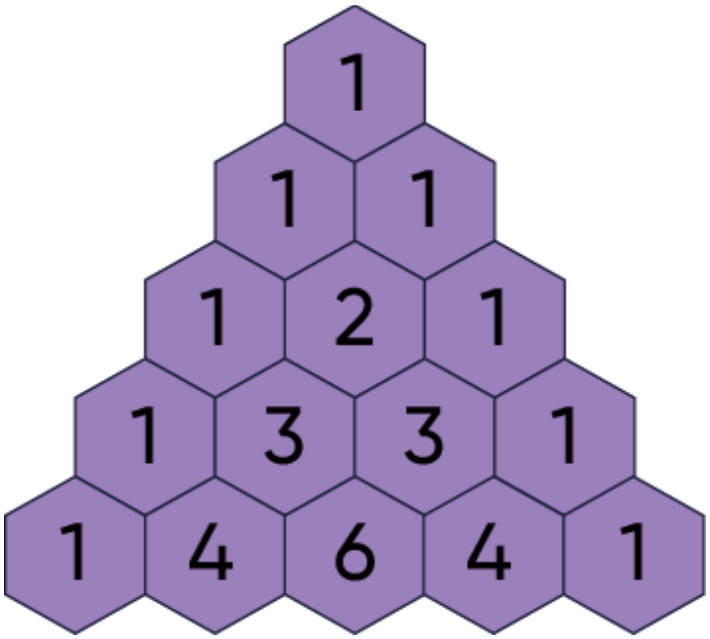
```
vector<int> v1(3, 10); // vector v1 {10, 10, 10}
vector<int> v2(3, 20); // vector v2 {20, 20, 20}
vector<int> v3(3, 3); // vector v3 {30, 30, 30}

vector<vector<int>> vect {v1, v2, v3};
```

Initializing a 2-D vector of n rows and m columns

```
vector<vector<int>> vect( n ,vector<int> (m));
```

Let us now solve some problem to test our understanding so far.

Problems
Q1. Given an integer n, return the first n rows of Pascal's triangle.
In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:



Example 1:
Input: n = 5
Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Example 2:
Input: n = 1
Output: [[1]]

Explanation of approach:

- Create an auxiliary 2D vector to store generated pascal triangle values having n rows.
- Iterate through every row and store integer(s) in it.
- Every row will have a number of columns equal to the row number.
- First and last column in every row are 1.
- Other values are the sum of values just above and left of above.

Let us now code the approach.

Solution:

```
#include <iostream>
#include<vector>
using namespace std;
vector<vector<int>> pascal(int numRows) {
    vector<vector<int>> ret;
    for (int i = 0; i < numRows; i++) {
        vector<int> row(i + 1, 1);
        for (int j = 1; j < i; j++) {
            row[j] = ret[i - 1][j] + ret[i - 1][j - 1];
        }
        ret.push_back(row);
    }
    return ret;
}
int main(){
    int n;
    cout<<"Enter number of rows you want : ";
    cin>>n;
    vector<vector<int>>ans;
    ans = pascal(n);
    for(int i=0;i<ans.size();i++){
        for(int j=0;j<ans[i].size();j++){
            cout<<ans[i][j]<<" ";
        }
        cout<<endl;
    }
    return 0;
}
```

OUTPUT :

```
Enter number of rows you want : 6
1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5  10  10  5  1
```

Upcoming Class Teasers

- 2D Array problems