LingoLeap Assignment Answer

Name: Sumit Rajkumar Bashetwar
mail:sumitbashetwar@gmail.com
Contact: 8975688691

```
#include <vector>
#include <iostream>

using namespace std;

int maxMoves(vector<vector<int>>& grid) {
  int m = grid.size();
  int n = grid[0].size();
  //initializing the vector of vector grid havin m rows and n columns to -1
  vector<vector<int>> dp(m, vector<int>(n, -1));

  // Initialize the base cases.
  for (int i = 0; i < m; i++) {
    dp[i][0] = 0;
  }

  // Iterate over the grid in bottom-up order.
  for (int i = m - 1; i >= 0; i--) {
    for (int j = 1; j < n; j++) {
      // Find the maximum number of moves that can be made from the current cell.
      int maxMoves = -1;
      for (int k = i - 1; k <= i + 1; k++) {
        if (k >= 0 && k < m && grid[k][j] > grid[i][j]) {
          maxMoves = max(maxMoves, dp[k][j] + 1);
        }
      }

      // Update the dp table.
      dp[i][j] = maxMoves;
    }
  }

  // Return the maximum number of moves that can be made from any cell in the first column.
  int maxMoves = -1;
  for (int i = 0; i < m; i++) {
    maxMoves = max(maxMoves, dp[i][0]);
  }
  return maxMoves;
}

int main() {
  // Get the input from the user.
  int m, n;
  cout << "Enter the number of rows: ";
```

```
  cin >> m;
  cout << "Enter the number of columns: ";
  cin >> n;

  vector<vector<int>> grid(m, vector<int>(n));
  for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
      cout << "Enter the value at row " << i << " and column " << j << ": ";
      cin >> grid[i][j];
    }
  }

  // Find the maximum number of moves that can be made.
  int max = maxMoves(grid);

  // Print the output.
  cout << "The maximum number of moves that can be made is: " << max << endl;
  return 0;
}
```