Print all possible paths from top left to bottom right of a mXn matrix

The problem is to print all the possible paths from top left to bottom right of a mXn matrix with the constraints that ***from each cell you can either move only to right or down***.

The algorithm is a simple recursive algorithm, from each cell first print all paths by going down and then print all paths by going right. Do this recursively for each cell encountered.

Following is C++ implementation of the above algorithm.

```
#include<iostream>
using namespace std;

/* mat:  Pointer to the starting of mXn matrix
   i, j: Current position of the robot (For the first call use 0,0)
   m, n: Dimentions of given the matrix
   pi:   Next index to be filed in path array
   *path[0..pi-1]: The path traversed by robot till now (Array to hold the
                   path need to have space for at least m+n elements) */
void printAllPathsUtil(int *mat, int i, int j, int m, int n, int *path, int pi)
{
    // Reached the bottom of the matrix so we are left with
    // only option to move right
    if (i == m - 1)
    {
        for (int k = j; k < n; k++)
            path[pi + k - j] = *((mat + i*n) + k);
        for (int l = 0; l < pi + n - j; l++)
            cout << path[l] << " ";
        cout << endl;
        return;
    }

    // Reached the right corner of the matrix we are left with
    // only the downward movement.
    if (j == n - 1)
    {
        for (int k = i; k < m; k++)
            path[pi + k - i] = *((mat + k*n) + j);
        for (int l = 0; l < pi + m - i; l++)
            cout << path[l] << " ";
        cout << endl;
        return;
    }

    // Add the current cell to the path being generated
    path[pi] = *((mat + i*n) + j);

    // Print all the paths that are possible after moving down
    printAllPathsUtil(mat, i+1, j, m, n, path, pi + 1);

    // Print all the paths that are possible after moving right
    printAllPathsUtil(mat, i, j+1, m, n, path, pi + 1);

    // Print all the paths that are possible after moving diagonal
    // printAllPathsUtil(mat, i+1, j+1, m, n, path, pi + 1);
}

// The main function that prints all paths from top left to bottom right
// in a matrix 'mat' of size mXn
void printAllPaths(int *mat, int m, int n)
{
    int *path = new int[m+n];
    printAllPathsUtil(mat, 0, 0, m, n, path, 0);
}

// Driver program to test abve functions
int main()
{
    int mat[2][3] = { {1, 2, 3}, {4, 5, 6} };
    printAllPaths(*mat, 2, 3);
    return 0;
}
```

Run on IDE

Output:

```
1 4 5 6
1 2 5 6
1 2 3 6
```

Note that in the above code, the last line of printAllPathsUtil() is commented, If we uncomment this line, we get all the paths from the top left to bottom right of a nXm matrix if the diagonal movements are also allowed. And also if moving to some of the cells are not permitted then the same code can be improved by passing the restriction array to the above function and that is left as an exercise.

This article is contributed by **Hariprasad NG**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above