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***.

Examples :

```
Input : 1 2 3
        4 5 6
Output : 1 4 5 6
         1 2 5 6
         1 2 3 6

Input : 1 2
        3 4
Output : 1 2 4
         1 3 4
```

```
def printPaths(matrix):
    ans = []
    n = len(matrix)
    m = len(matrix[0])
    visited = [[False] * m for i in range(n)]
    printPathsUtil(matrix, ans, 0, 0, n, m, visited, '')
    return ans


def printPathsUtil(matrix, ans, i, j, n, m, visited, ssf):
    if i == n or j == m:
        return
    if i == n - 1 and j == m - 1:
        ans.append(ssf)
        return
    if visited[i][j] is False:
        visited[i][j] = True
        printPathsUtil(matrix, ans, i, j + 1, n, m, visited, ssf +
str(matrix[i][j]))
        printPathsUtil(matrix, ans, i + 1, j, n, m, visited, ssf +
str(matrix[i][j]))
```

```
        visited[i][j] = False


matrix = [[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]]
print(printPaths(matrix))
```

['1236', '1256', '1258', '1456', '1458', '1478']

Just add last element to the elements and you have your answer