Print Minimum Gold Path

```
import sys


def minimumGoldPath(matrix):
    n = len(matrix)
    m = len(matrix[0])
    dp = [[0] * m for _ in range(n)]
    for i in range(n - 1, -1, -1):
        for j in range(m - 1, -1, -1):
            if i == n - 1 and j == m - 1:
                dp[i][j] = matrix[i][j]
            elif i == n - 1:
                dp[i][j] = matrix[i][j] + dp[i][j + 1]
            elif j == m - 1:
                dp[i][j] = matrix[i][j] + dp[i + 1][j]
            else:
                dp[i][j] = matrix[i][j] + min(dp[i][j + 1], dp[i + 1][j])
    # print(dp)
    queue = []
    var = ('S===>', 0, 0)
    queue.append(var)
    while len(queue):
        ssf, idx, idy = queue.pop(0)
        if idx == n - 1 and idy == m - 1:
            print(ssf)
        else:
            if idx == n - 1:
                var = (ssf + 'H', idx, idy + 1)
                queue.append(var)
            elif idy == m - 1:
                var = (ssf + 'V', idx + 1, idy)
                queue.append(var)
            else:
                if dp[idx][idy + 1] > dp[idx + 1][idy]:
                    var = (ssf + 'V', idx + 1, idy)
                    queue.append(var)
                elif dp[idx][idy + 1] < dp[idx + 1][idy]:
                    var = (ssf + 'H', idx, idy + 1)
                    queue.append(var)
```

```
                else:
                    queue.append((ssf + 'H', idx, idy + 1))
                    queue.append((ssf + 'V', idx + 1, idy))


matrix = [[0, 1, 4, 2, 8, 2],
          [4, 3, 6, 5, 0, 4],
          [1, 2, 4, 1, 4, 6],
          [2, 0, 7, 3, 2, 2],
          [3, 1, 5, 9, 2, 4],
          [2, 7, 0, 8, 5, 1]]
minimumGoldPath(matrix)
```