Problem C

-If you are using adjacency matrix check adj[i][k] == adj[i][j] == adj[j][k]

Problem E

Problem F

-Use DFS or BFS algorithm to find path

-For every v store pv – the vertex you came to v from

-Traverse reversed path using pv

-Don’t use std::endl, whe outputing many lines, because it flushes output every time you call it

Problem G

-Dynamic programming: f[i][j] – number of ways to reach cell (i, j) from (1, 1)

-Formula: f[i][j] = f[i – 1][j – 2] + f[i – 2][j – 1]

-Output f[n][m]

-For f[1][1] = 1

Problem H

-Get all pairs of numbers having the same number of digits

-Find their longest common prefix

-acdj and acdf has common prefix of 3, if acdj is less tan acdf, then we know that letter j is less tan letter f

-Build a graph, where letters are vértices, and there is an edge, if one letter is earlier in alphabetic order tan another

-Topological sorting of this graph is an answer