This exercise covers some aspects of Johnson's paper on *yacc*[1] along with some graph terminology, both of which are featured in the next homework.

1. List the members of your group below:

2. Earlier exercises have introduced several well-known graphs, such as K_n and C_n. Another such graph is a $m \times n$ *grid graph* (or *lattice graph*, or *mesh graph*) $G_{m,n} = (V, E)$ where $V = \{(x, y) \mid x \in 0, 1, 2, \ldots, m \text{ and } y \in 0, 1, 2, \ldots, n\}$ and $E = \{(p, q) \mid p, q \in V \text{ and } d(p, q) = 1\}$, using $d(p, q)$ to denote the Euclidean distance between points p and q.

   Depict $G_{m,n}$ for $0 \leq n \leq m \leq 5$.

[1] Stephen C. Johnson, Yacc: Yet Another Compiler-Compiler, Unix Programmer's Manual, Volume 2b. AT&T Bell Laboratories. Murray Hill, New Jersey, 1978.

3. Is it possible to generate $G_{m,n}$ (Question 2) using the *graph calculator* described in the homework assignment? If so, provide the calculator input that generates and prints the graph; else explain why not.

4. Determine the number of paths from the origin $(0, 0)$ of a grid graph to the vertex (m, n) assuming all edges are oriented away from the origin. Justify your answer. Present the numerical answers for $0 \leq n \leq m \leq 5$.

5. Determine the number of paths from $(0, 0)$ to (m, n), as in Question 4, but subject to the constraint that the paths are not permitted to visit any vertex (x, y) with $x < y$. Present the numerical answers for $0 \leq n \leq m \leq 5$. Comment on any observed similarities to previously studied sequences.

6. Recall the question from the midterm exam that asked for a trace of bit-splitting radix sort. Provide a *bash* script that generates the answer to that question given the data (in decimal) as input. The goal is to generate as concise a script as possible. You may use any of the standard Unix tools, such as *grep*, *sed*, *awk*, *sort*, etc. (Ask for clarifications about tool use.)