1. (5 points) **Finding Shortest Paths**

Given an \( m \times n \) matrix of integers, structure a program that computes a path of minimal weight. The weight of a path is the sum of the integers in each of the \( n \) cells of the matrix that are visited. A path starts anywhere in column 1 (the first column) and consists of a sequence of steps terminating in column \( n \) (the last column). A step consists of traveling from column \( i \) to column \( i + 1 \) in an adjacent (horizontal or diagonal) row. For example, a \( 5 \times 6 \) matrix and its shortest path are shown below:

\[
\begin{array}{cccccc}
5 & 4 & 1 & 2 & 8 & 6 \\
6 & 1 & 8 & 2 & 7 & 4 \\
3 & 9 & 3 & 9 & 9 & 5 \\
8 & 4 & 1 & 3 & 2 & 6 \\
3 & 7 & 2 & 8 & 6 & 4 \\
\end{array}
\]
2. (5 points) **Activity Selection**

In the *Activity Selection* problem, we wish to select a maximum-size subset of mutually compatible activities. We know that ordering by the *finish times* produces an *optimal* solution.

1. Show an input for which ordering by the *starting times* *does not* produce an optimal solution.
2. Show an input for which ordering by *length* (shortest to longest) *does not* produce an optimal solution.
3. The *degree* of an activity is the number of activities whose time intervals intersect with it. Show an input for which ordering by degree (smallest to largest) *does not* produce an optimal solution.
3. (5 points) **Huffman Code**

   Use **Huffman coding** to encode these symbols with given frequencies:

   A: 0.10, B: 0.25, C: 0.05, D: 0.15, E: 0.30, F: 0.07, G: 0.08.

   Which data structure can be used for Huffman coding? What is the total running time of Huffman coding on a set of $n$ characters?
4. (5 points) **K Bishops**

We have previously solved the **N Queens** Problem, where, for a given \( n \), we calculated the number of ways to place \( n \) queens on an \( n \times n \) board. This problem concerns **bishops** on the chessboard. A bishop is a chess piece that controls all the squares on the two diagonals that it can reach. The **K Bishops** Problem should calculate the number of different ways that \( k \) bishops could be placed on an \( n \times n \) chessboard. Structure your program using the backtracking scheme that we have used for the **N Queens**. What needs to be modified (mostly) is the **OK** function.

Write the `bool isOK(...)` function for **K Bishops** problem.
5. (5 points) **Graph Representations**

1. What is the advantage and disadvantage of using an **adjacency-list** representation?
2. What is the advantage and disadvantage of using an **adjacency-matrix** representation?
3. Draw the two representations of the graph below:
6. (5 points) **Graphic Sequence**

Use **Graphic Sequence Transformation** to show that sequence \((2, 4, 2, 5, 2, 4, 5, 2)\) is graphic or not.