You are allowed to discuss with others but are not allowed to use any references other than the course notes and the three reference books. Please list your collaborators for each question. You must write your own solutions. See the course outline for the homework policy.

There are totally 55 marks (including the bonus). The full mark is 50 (extra marks above 50 will not be carried over). This homework is counted 8% of the course grade.

---

1. **Programming Problem: Marble Game** (16 marks)

   The instructions for submitting your programs will be posted on piazza.

   Consider the following game, which is played on a long board and with four marbles. The board consists of $n \geq 4$ marble slots arranged in a row, which we number from 1 to n, from left to right. Each marble fits in one of the slots.

   A game state X is represented using a four-tuple (X_1, X_2, X_3, X_4), where $1 \leq X_1 < X_2 < X_3 < X_4 \leq n$ are the locations of the marbles. The marbles are identical, so swapping two marbles does not change the state.

   There are two types of moves that one can make. The first type is to move a marble to an adjacent, unoccupied slot. The second type is to take a marble and "reflect" it along another marble. The requirements are that the distance between them is maintained, and there is no third marble along the way. Formally, if we take a marble at slot p, we can reflect it along a marble at slot q, bringing it to slot $2q - p$, as long as there are no other marbles between slots p and $2q - p$ (inclusive). The following figure depicts situations where a move can and cannot be made.

Figure 1: On the top is the current state, encoded $(2, 3, 5, 8)$. By moving the marble at position 2 to the left, we can reach state $(1, 3, 5, 8)$. By reflecting the marble at position 2 along the marble at position 3, we can reach state $(3, 4, 5, 8)$. By reflecting the marble at position 3 along the marble at position 5, we can reach state $(2, 5, 7, 8)$. We cannot reflect the marble at position 5 along the marble at position 3, because the marble at position 2 gets in the way.

Your task is to find a shortest sequence of moves from a starting state S to a target state T.

Input: The first line has an integer n, representing the length of the board. The second line has four integers $S_1, \ldots, S_4$, representing the starting state $S = (S_1, S_2, S_3, S_4)$. The third line has four integers $T_1, \ldots, T_4$, representing the target state $T = (T_1, T_2, T_3, T_4)$.

It is guaranteed that $4 \leq n \leq 80$, $1 \leq S_1 < S_2 < S_3 < S_4 \leq n$ and $1 \leq T_1 < T_2 < T_3 < T_4 \leq n$.

Output: The first line should be the minimum number of moves M required to teach the target state from the starting state. $M+1$ lines should follow. The i-th of these lines should be $X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)}$, representing the i-th state $X^{(i)} = (X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)})$ in the sequence of moves. The output should be so that:

- $X^{(1)} = S$ and $X^{(M+1)} = T$;
- State $X^{(i+1)}$ is reachable from state $X^{(i)}$ in one move, for all $1 \leq i \leq M$;
- $X_1^{(i)} < X_2^{(i)} < X_3^{(i)} < X_4^{(i)}$ for all $1 \leq i \leq M+1$.

If there are multiple shortest sequences of moves, any of them will be accepted.

Sample Input:

```
5
1 2 3 4
2 3 4 5
```

Sample Output:

```
2
1 2 3 4
1 2 4 5
2 3 4 5
```

2. **Written Problem: Number of Shortest s-t Paths** (12 marks)

   Given an undirected graph $G = (V, E)$ and two vertices s and t, design an algorithm to return the number of different shortest s-t paths between s and t. Two paths are the same if they have exactly the same sequence of vertices; otherwise they are different.

   You will get full marks if the time complexity of the algorithm is $O(|V| + |E|)$ word operations and the proofs are correct. We assume that the answer fits in one word, so that adding two numbers with the sum being at most the answer can be done in one word operation.

3. **Written Problem: One-Way Streets** (12 marks)

Imagine in a city where all the streets are two-way streets. Perhaps because the streets are too narrow, the government would like to see if it is possible to make all streets one-way, so that there is still a way to drive from any intersection to any other intersection in the city.

Show how to model this problem as a graph problem. Then design an algorithm to determine if it is possible, and when it is possible to output one solution to assign the directions.

You will get full marks if the time complexity of the algorithm is $O(n+m)$ and the proofs are correct, where m is the number of streets in the city and n is the number of intersections in the city.

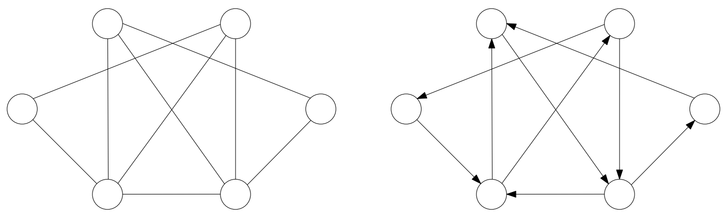


Figure 2: There is a solution in this example.

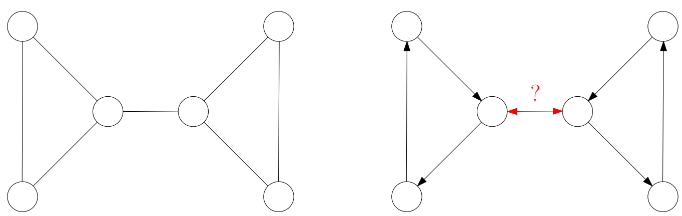
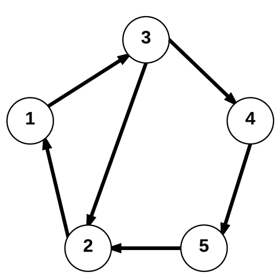


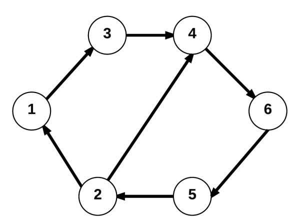
Figure 3: There is no solution in this example.

4. **Written Problem: Directed Odd Cycle** (12+3 marks)

Given a directed graph $G = (V, E)$, design an algorithm to determine if there is a directed odd cycle in G or not. You will get full marks if the time complexity is $O(|V| + |E|)$ and the proofs are correct.



(a) A graph with a directed odd cycle $1 \to 3 \to 2 \to 1$.



(b) A graph without directed odd cycles.

Bonus: (3 marks) You will get up to 3 bonus marks if your algorithm can print out a simple directed odd cycle when it exists.