Problem Solving & Coding – Level I
[Data Structure Foundations]
(Array Applications)

Problem 1: Largest Palindromic Number

You are given a string *num* consisting of digits only. Return the largest palindromic integer (in the form of a string) that can be formed using digits taken from *num*. It should not contain leading zeroes.

Input: num = "444947137"
Output: "7449447"
Source: https://leetcode.com/problems/largest-palindromic-number/

Problem 2: Sudoku Verifier

Determine if a 9 x 9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:

1. Each row must contain the digits 1-9 without repetition.
2. Each column must contain the digits 1-9 without repetition.
3. Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

Input: board =

```
[["5","3",".",".","7",".",".",".","."]
,["6",".",".","1","9","5",".",".","."]
,[".","9","8",".",".",".",".",".","6","."]
,["8",".",".",".","6",".",".",".","3"]
,["4",".",".","8",".","3",".",".","1"]
,["7",".",".",".","2",".",".",".","6"]
,[".","6",".",".",".",".",".","2","8","."]
,[".",".",".","4","1","9",".",".","5"]
,[".",".",".",".","8",".",".","7","9"]]
```

Output: true
Source: https://leetcode.com/problems/valid-sudoku/description/

Problem 3: Faster Prime Generator

Given an integer n, return the number of prime numbers that are strictly less than n. You must use the following algorithm, called as Sieve method.

Write down all of the numbers from 0 to the limit
Cross out 0 and 1 because they are not prime

Set *p* equal to 2
While *p* is less than the limit **do**
  Cross out all multiples of *p* (but not *p* itself)
  Set *p* equal to the next number in the list that is not crossed out

Report all of the numbers that have not been crossed out as prime

Input: n = 10
Output: 4
Explanation: There are 4 prime numbers less than 10, they are 2, 3, 5, 7.
Source: https://leetcode.com/problems/count-primes/description/

Problem4: BigInteger Multiplication

Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2, also represented as a string. You must not use any built-in BigInteger library or convert the inputs to integer directly.

Input: num1 = "123", num2 = "456"
Output: "56088"
Source: https://leetcode.com/problems/multiply-strings/

Problem 5: Card Game

A standard deck of playing cards contains 52 cards. Each card has one of four suits along with a value. The suits are normally spades, hearts, diamonds and clubs while the values are 2 through 10, Jack, Queen, King and Ace. Each playing card can be represented using two characters. The first character is the value of the card, with the values 2 through 9 being represented directly. The characters “T”, “J”, “Q”, “K” and “A” are used to represent the values 10, Jack, Queen, King and Ace respectively. The second character is used to represent the suit of the card. It is normally a lowercase letter: “s” for spades, “h” for hearts, “d” for diamonds and “c” for clubs. The following table provides several examples of cards and their two-character representations:

Problem Solving & Coding – Level I
[Data Structure Foundations]
(Array Applications)

Card	Abbreviation
Jack of spades	Js
Two of clubs	2c
Ten of diamonds	Td
Ace of hearts	Ah
Nine of spades	9s

Create a function named *createDeck*. It will use loops to create a complete deck of cards by storing the two-character abbreviations for all 52 cards into an array. Return the array of cards as the function's only result. Your function will not take any input parameters.

Create another function named *shuffleDeck* that randomizes the order of the cards in an array. One technique that can be used to shuffle the cards is to visit each element in the list and swap it with another random element in the list. You must write your own loop for shuffling the cards. You cannot make use of any built-in shuffle function.

Write a main program that displays the cards before and after shuffle. Then, it must take number of hands n as input and prints `n` poker hands (five cards each) from a shuffled deck, separated by blank lines.