Lab 5C

Problem 1: Largest Number

Description:

Given a list of non-negative integers `nums` , arrange them such that they form the largest number and return it.

Since the result may be very large, you need to return a string instead of an integer.

Input Format:

```
The input consists of  'n' and line containing the number of elements followed by the elements of the array:
Input: n
    nums[0] nums[1] ... nums[n-1]
```

Output Format:

```
The output is a single string representing the largest number that can be formed.
Output:
    largest_number_string
```

Constraints:

- `1 <= nums.length (n) <= 100`
- `0 <= nums[i] <= 10^9`

Example 1:

```
Input:  2
        10 2

Output: 210
```

Example 2:

```
Input:  5
        3 30 34 5 9

Output: 9534330
```

Problem 2: Predict the Winner

Description:

You are given an integer array `nums` . Two players are playing a game with this array: player 1 and player 2.

Player 1 and player 2 take turns, with player 1 starting first. Both players start the game with a score of 0. At each turn, the player takes one of the numbers from either end of the array (i.e., `nums[0]` or `nums[nums.length - 1]`) which reduces the size of the array by 1. The player adds the chosen number to their score. The game ends when there are no more elements in the array.

Return `true` if Player 1 can win the game. If the scores of both players are equal, then player 1 is still the winner, and you should also return `true` . You may assume that both players are playing optimally.

Input Format:

```
The input consists of a single line containing the number of elements followed by the elements of the array:
Input: n
        nums[0] nums[1] ... nums[n-1]
```

Output Format:

```
The output is a boolean value indicating whether Player 1 can win.
Output:
        true or false
```

Constraints:

- `1 <= nums.length (n) <= 20`
- `0 <= nums[i] <= 10^7`

Example 1:

```
Input: 3
        1 5 2
Output:
        false
```

Example 2:

```
Input: 4
        1 5 233 7
Output:
        true
```

Problem 3: Zigzag Conversion

Description:

The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P   A   H   N
A P L S I I G
Y   I   R
```

And then read line by line: "PAHNAPLSIIGYIR".

Write the code that will take a string and make this conversion given a number of rows:

Input Format:

```
The input consists of two lines:
- The first line contains the string s.
- The second line contains the integer numRows.
Input:
        s
        numRows
```

Output Format:

```
The output is a single string representing the zigzag conversion.
Output:
        zigzag_converted_string
```

Constraints:

- `1 <= s.length<= 1000`
- `s` consists of English letters (lower-case and upper-case), ',' and '.'
- `1 <= numRows <= 1000`

Example 1:

```
Input:
        PAYPALISHIRING
        3
Output:
        PAHNAPLSIIGYIR
```

Example 2:

```
Input:
        PAYPALISHIRING
        4
Output: PINALSIGYAHRPI
```

Example 3:

```
Input:
        A
        1
Output:
        A
```

Submission Guidelines

Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.