EXAM 姓名 _____

- Can use any programming language.
- Answer any 3 questions. If you still have time can do more.
- No consider build failed or exception.
- No advanced library using.
- No discussion.
- No open book.

---

1. Two Sum

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

// Code

```
int[] TwoSum(int[] nums, int target)
{
    // TODO: your implement
}
```

2. Exist in an array

Given an array without sorting `nums` and a number `n`, return *true or false to indicate whether* `n` *is in* `nums` *or not.*

Example 1:

```
Input: nums = [3,13,1,5,5,14,17,78], n = 14
Output: true
```

Example 2:

```
Input: nums = [3,2,4], n = 6
Output: false
```

```
// Code
bool IsExistInArray(int[] nums, int n)
{
    // TODO: your implement
}
```

3. Search Insert Position

Given a sorted ascending array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

Example 1:

```
Input: nums = [1,3,5,6], target = 5
Output: 2
```

Example 2:

```
Input: nums = [1,3,5,6], target = 2
Output: 1
```

Example 3:

```
Input: nums = [1,3,5,6], target = 7
Output: 4
```

// Code

```
int SearchInsert(int[] nums, int target)
{
    // TODO: your implement
}
```

4. Reverse String

Write a function that reverses a string. The input string is given as an array of characters `s`.

You must do this by modifying the input array in-place.

Example **1:**

```
Input: s = ["h","e","l","l","o"]
Output: ["o","l","l","e","h"]
```

Example 2:

```
Input: s = ["H","a","n","n","a","h"]
Output: ["h","a","n","n","a","H"]
```

// Code

```
void ReverseString(char[] s)
{
    // TODO: your implement
}
```

5. Combine Two Tables

Write an SQL query to report the first name, last name, city, and state of each person in the `Person` table. If the address of a `personId` is not present in the `Address` table, report `null` instead.

Return the result table in **any order**.

The query result format is in the following example.

Example 1:

```
Input:
Person table:
+----------+----------+-----------+
| personId | lastName | firstName |
+----------+----------+-----------+
| 1        | Wang     | Allen     |
| 2        | Alice    | Bob       |
+----------+----------+-----------+
Address table:
+-----------+----------+---------------+------------+
| addressId | personId | city          | state      |
+-----------+----------+---------------+------------+
| 1         | 2        | New York City | New York   |
| 2         | 3        | Leetcode      | California |
+-----------+----------+---------------+------------+
Output:
+-----------+----------+---------------+----------+
| firstName | lastName | city          | state    |
+-----------+----------+---------------+----------+
| Allen     | Wang     | Null          | Null     |
| Bob       | Alice    | New York City | New York |
+-----------+----------+---------------+----------+
Explanation:
There is no address in the address table for the personId = 1 so we return
null in their city and state.
addressId = 1 contains information about the address of personId = 2.
```

6. OOP

Please briefly describe the three major characteristics of object-oriented and a description of their characteristics.