

## C强化第一讲

1. 实现函数 `int sqrt(int x)`，计算并返回  $x$  的平方根（向下取整）

**要求：**空间复杂度  $O(1)$ ，时间复杂度  $O(\log x)$

**示例：**

输入：2  
返回值：1

2. 有一个长度为  $n$  的按严格升序排列的整数数组 `nums`，在实行 `search` 函数之前，在某个下标  $k$  上进行旋转，使数组变为 `[nums[k], nums[k+1], ..., nums[nums.length-1], nums[0], nums[1], ..., nums[k-1]]`。

给定旋转后的数组 `nums` 和一个整型 `target`，请你查找 `target` 是否存在于 `nums` 数组中并返回其下标（从0开始计数），如果不存在请返回-1。

比如，数组 `[0,2,4,6,8,10]` 在下标3处旋转之后变为 `[6,8,10,0,2,4]`，当给定 `target` 为10时，10的下标是2，`target` 为3时，`nums` 数组中不存在3，所以返回-1

**要求：**空间复杂度  $O(1)$ ，时间复杂度  $O(\log n)$

**示例：**

输入：`[6,8,10,0,2,4]`，10  
返回值：2

3. 给你一个大小为  $n$  的字符串数组 `strs`，其中包含  $n$  个字符串，编写一个函数来查找字符串数组中的最长公共前缀，返回这个公共前缀。

**示例：**

输入：`["abca", "abc", "abca", "abc", "abcc"]`  
返回值：`"abc"`

4. 给定有符号整数 `num`，将 `num` 中的数字部分反转，最后返回反转的结果，只反转数字部分，符号位部分不反转

**示例:**

输入: -123  
返回值: -321

5. 给定一个二进制01串, 输出它的十进制形式。

**示例:**

输入: 110  
返回值: 6

6. 对于一个字符串, 我们需要对它做一些变形。

首先这个字符串中包含着一些空格, 就像"Hello World"一样, 然后我们要做的是把这个字符串中由空格隔开的单词反序, 同时反转每个字符的大小写。

比如"Hello World"变形后就变成了"wORLD hELLO"。字符串中包括大写英文字母、小写英文字母、空格。

**要求:** 时间复杂度  $O(n)$

**示例:**

输入: "This is a sample"  
返回值: "SAMPLE A IS tHIS"

7. 有一个长度为  $n$  的非降序数组, 比如[1,2,3,4,5], 将它进行旋转, 即把一个数组最开始的若干个元素搬到数组的末尾, 变成一个旋转数组, 比如变成了[3,4,5,1,2], 或者[4,5,1,2,3]这样的。请问, 给定这样一个旋转数组, 求数组中的最小值。

**要求:** 空间复杂度:  $O(1)$ , 时间复杂度:  $O(\log n)$

**示例:**

输入: [3,4,5,1,2]  
返回值: 1

8. 给定一个长度为  $n$  的非降序数组和一个非负数整数  $k$ ，要求统计  $k$  在数组中出现的次数

**要求：**空间复杂度  $O(1)$ ，时间复杂度  $O(\log n)$

**示例：**

输入: `[1,2,3,3,3,3,4,5]`, `8,3`  
返回值: `4`

9. 给定一个数组，请你实现将所有 0 移动到数组末尾并且不改变其他数字的相对顺序。

**要求：**时间复杂度为  $O(n)$ ，空间复杂度  $O(1)$

**示例：**

输入: `[1,2,0,3]`, `4`  
返回值: `[1,2,3,0]`

10. 以字符串的形式读入两个数字，编写一个函数计算它们的和，以字符串形式返回，字符串仅由 '0'~'9' 构成

**要求：**时间复杂度  $O(n)$

**示例：**

输入: `"1"`, `"99"`  
返回值: `"100"`