学习任务

Task01：数组（1天）

理论部分

- 理解数组的存储与分类。
- 实现动态数组，该数组能够根据需要修改数组的长度。

练习部分

1. 利用动态数组解决数据存放问题

编写一段代码，要求输入一个整数N，用动态数组A来存放2~N之间所有5或7的倍数，输出该数组。

示例：

```
输入：
N = 100

输出：
5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50 55 56 60 63 65 70 75 77 80 84
85 90 91 95 98 100
```

2. 托普利茨矩阵问题

如果一个矩阵的每一方向由左上到右下的对角线上具有相同元素，那么这个矩阵是托普利茨矩阵。

给定一个M x N的矩阵，当且仅当它是托普利茨矩阵时返回True。

示例：

```
输入：
matrix = [
  [1,2,3,4],
  [5,1,2,3],
  [9,5,1,2]
]

输出: True
```

解释:

在上述矩阵中, 其对角线为: "[9]", "[5, 5]", "[1, 1, 1]", "[2, 2, 2]", "[3, 3]", "[4]"。
各条对角线上的所有元素均相同, 因此答案是True。

3. 三数之和

https://leetcode-cn.com/problems/3sum/

给定一个包含 n 个整数的数组`nums`，判断`nums`中是否存在三个元素`a，b，c`，使得`a + b + c = 0`？找出所有满足条件且不重复的三元组。

注意：答案中不可以包含重复的三元组。

示例：

```
给定数组 nums = [-1, 0, 1, 2, -1, -4],

满足要求的三元组集合为:
[
  [-1, 0, 1],
  [-1, -1, 2]
]
```

Task02：顺序表和链表（2天）

理论部分

- 理解线性表的定义与操作。
- 实现顺序表。
- 实现单链表、循环链表、双向链表。

练习部分

1. 合并两个有序链表

https://leetcode-cn.com/problems/merge-two-sorted-lists/

将两个有序链表合并为一个新的有序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。

示例：

```
输入：1->2->4, 1->3->4
输出：1->1->2->3->4->4
```

2. 删除链表的倒数第**N**个节点

https://leetcode-cn.com/problems/remove-nth-node-from-end-of-list/

给定一个链表，删除链表的倒数第 *n* 个节点，并且返回链表的头结点。

示例：

```
给定一个链表: 1->2->3->4->5, 和 n = 2.

当删除了倒数第二个节点后，链表变为 1->2->3->5.
```

说明：

给定的 n 保证是有效的。

进阶：

你能尝试使用一趟扫描实现吗？

3. 旋转链表

https://leetcode-cn.com/problems/rotate-list/

给定一个链表，旋转链表，将链表每个节点向右移动*k*个位置，其中*k*是非负数。

示例 1:

```
输入: 1->2->3->4->5->NULL, k = 2
输出: 4->5->1->2->3->NULL

解释:
向右旋转 1 步: 5->1->2->3->4->NULL
向右旋转 2 步: 4->5->1->2->3->NULL
```

示例 2:

```
输入: 0->1->2->NULL, k = 4
输出: 2->0->1->NULL

解释:
向右旋转 1 步: 2->0->1->NULL
向右旋转 2 步: 1->2->0->NULL
向右旋转 3 步: 0->1->2->NULL
向右旋转 4 步: 2->0->1->NULL
```

Task03：栈与递归（2天）

理论部分

- 用数组实现一个顺序栈。
- 用链表实现一个链栈。
- 理解递归的原理。

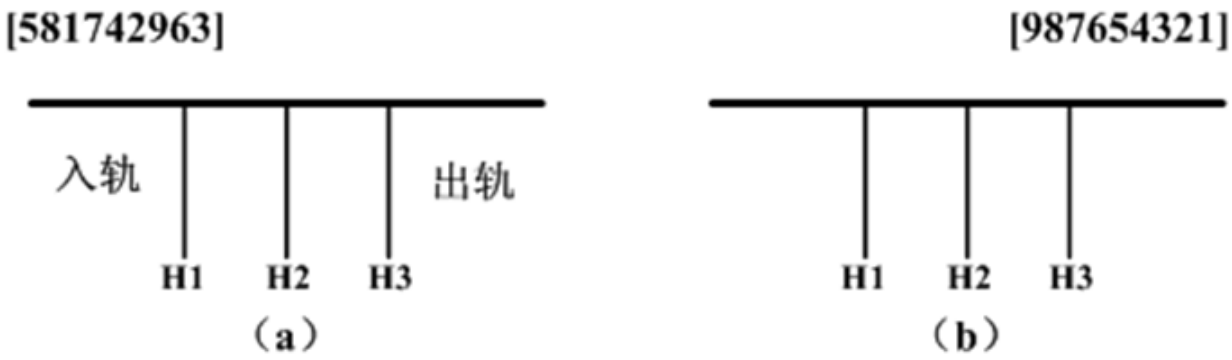
练习部分

1. 根据要求完成车辆重排的程序代码

假设一列货运列车共有n节车厢，每节车厢将停放在不同的车站。假定n个车站的编号分别为1至n，货运列车按照第n站至第1站的次序经过这些车站。车厢的编号与它们的目的地相同。为了便于从列车上卸掉相应的车厢，

必须重新排列车厢，使各车厢从前至后按编号1至n的次序排列。当所有的车厢都按照这种次序排列时，在每个车站只需卸掉最后一节车厢即可。

我们在一个转轨站里完成车厢的重排工作，在转轨站中有一个入轨、一个出轨和k个缓冲铁轨（位于入轨和出轨之间）。图（a）给出一个转轨站，其中有k个（k=3）缓冲铁轨H1，H2 和H3。开始时，n节车厢的货车从入轨处进入转轨站，转轨结束时各车厢从右到左按照编号1至n的次序离开转轨站（通过出轨处）。在图（a）中，n=9，车厢从后至前的初始次序为5，8，1，7，4，2，9，6，3。图（b）给出了按所要求的次序重新排列后的结果。



编写算法实现火车车厢的重排，模拟具有n节车厢的火车“入轨”和“出轨”过程。

Task04：队列（2天）

理论部分

- 用数组实现一个顺序队列。
- 用数组实现一个循环队列。
- 用链表实现一个链式队列。

练习部分

1. 模拟银行服务完成程序代码。

目前，在以银行营业大厅为代表的窗口行业中大量使用排队（叫号）系统，该系统完全模拟了人群排队全过程，通过取票进队、排队等待、叫号服务等功能，代替了人们站队的辛苦。

排队叫号软件的具体操作流程为：

- 顾客取服务序号

当顾客抵达服务大厅时，前往放置在入口处旁的取号机，并按一下其上的相应服务按钮，取号机会自动打印出一张服务单。单上显示服务号及该服务号前面正在等待服务的人数。

- 服务员工呼叫顾客 服务员工只需按一下其柜台上呼叫器的相应按钮，则顾客的服务号就会按顺序的显示在显示屏上，并发出“叮咚”和相关语音信息，提示顾客前往该窗口办事。当一位顾客办事完毕后，柜台服务员工只需按呼叫器相应键，即可自动呼叫下一位顾客。

编写程序模拟上面的工作过程，主要要求如下：

- 程序运行后，当看到“请点击触摸屏获取号码：”的提示时，只要按回车键，即可显示“您的号码是：XXX，您前面有YYY位”的提示，其中XXX是所获得的服务号码，YYY是在XXX之前来到的正在等待服务的

人数。

- 用多线程技术模拟服务窗口（可模拟多个），具有服务员呼叫顾客的行为，假设每个顾客服务的时间是10000ms，时间到后，显示“请XXX号到ZZZ号窗口！”的提示。其中ZZZ是即将为客户服务的窗口号。

Task05：字符串（2天）

理论部分

- 用数组实现一个顺序的串结构。
- 为该串结构提供丰富的操作，比如插入子串、在指定位置移除给定长度的子串、在指定位置取子串、连接串、串匹配等。

练习部分

1. 题目

给定一个字符串，请你找出其中不含有重复字符的最长子串的长度。

```
输入: "abcabcbb"

输出: 3
```

解释: 因为无重复字符的最长子串是 "abc"，所以其长度为 3。

2. 题目

给定一个字符串s和一些长度相同的单词 words。找出 s 中恰好可以由 words 中所有单词串联形成的子串的起始位置。

注意子串要与 words 中的单词完全匹配，中间不能有其他字符，但不需要考虑 words 中单词串联的顺序。

```
输入:
  s = "barfoothefoobarman",
  words = ["foo","bar"]
输出: [0,9]
```

解释：

从索引 0 和 9 开始的子串分别是 "barfoo" 和 "foobar" 。输出的顺序不重要, [9,0] 也是有效答案。

3. 替换子串得到平衡字符串

https://leetcode-cn.com/problems/replace-the-substring-for-balanced-string/

有一个只含有'Q', 'W', 'E','R'四种字符，且长度为 n的字符串。假如在该字符串中，这四个字符都恰好出现n/4次，那么它就是一个「平衡字符串」。

给你一个这样的字符串 s，请通过「替换一个子串」的方式，使原字符串 s变成一个「平衡字符串」。你可以用和「待替换子串」长度相同的任何其他字符串来完成替换。

请返回待替换子串的最小可能长度。

如果原字符串自身就是一个平衡字符串，则返回 0。

示例1：

```
输入：s = "QWER"
输出：0
解释：s 已经是平衡的了。
```

示例2：

```
输入：s = "QQWE"
输出：1
解释：我们需要把一个 'Q' 替换成 'R'，这样得到的 "RQWE" (或 "QRWE") 是平衡的。
```

示例3：

```
输入：s = "QQQW"
输出：2
解释：我们可以把前面的 "QQ" 替换成 "ER"。
```

示例4：

```
输入：s = "QQQQ"
输出：3
解释：我们可以替换后 3 个 'Q'，使 s = "QWER"。
```