程序设计实践（二）收获与反思

今天主要学习了动态获取内存单元的单链表操作，相对于数组，其显著的特点是可以非常快捷的添加和删除内容，但是它同时也有劣势——需要通过遍历才能获得特定位置下的数据。在进行添加和删除操作时，为了避免操作位置越界，可以通过一个临时指针遍历一遍链表，获取链表的长度，当然，本人水平有限，目前只能找到这种拙劣的方法。

几个关键步奏：

1. 引入头文件 #include <malloc.h>
2. 请求内存单元 p = (struct Node \*)malloc(sizeof(struct Node))；
3. 链表单元连接操作p->next = head->next; head->next = p;（注意先后顺序）

注意这种带头结点的单链表生成的为倒序排列，在实际运用中需要根据具体情况，创建正序排列的链表，正序的创建方法：

temp = (struct Student \*)malloc(LEN);

temp->next = NULL;

p->next = temp;

p = temp;

需要注意先后顺序，并且这只是部分代码，需要灵活运用，可以先把一个head指针赋值为头结点，方便之后的操作。

另外，对螺旋矩阵也有一些理解，之前只能用笨方法硬求，经过思考，发现，递归的方法会更方便。

最后，温习了昨天的大数据处理的知识，对于阶乘运算，与N次方运算类似，只是将固定的数字变化了起来。

2019-7-5