**数据结构实验报告**

姓名： 粟锦 学号：U201817053 班级：软工1803班

**实验三**

**实验描述**

1. 约瑟夫环问题：用游标方式的循环链表的方式实现Josepuhs(n,m)问题的求解过程。
2. 多项式乘法。用链表表示多项式，分别在对指数排序和不排序的情况下，写出求两个给定多项式的乘法的函数。其计算复杂度分别是多少？

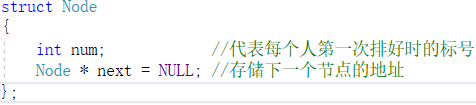
**约瑟夫环问题：**

分析：用游标的方式代替循环链表实现约瑟夫环，首先需要将需要构建出一个结构体数组充当存储空间，然后将存储空间以游标的方式进行初始化。最后根据用户输入的人数值进行游标循环的初始化，然后再进行对数组的计数遍历，依次淘汰数组中的元素直至最后只剩下一个元素，即为剩下的人。

求解过程：

构建游标的框架：

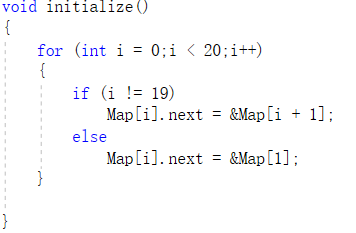
首先定义一个结构体，结构体中包含一个人的编号以及该结构体的一个指针来指向下一个人。



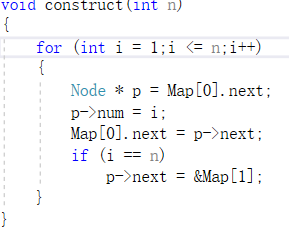
然后定义一个全局变量结构体数组充当存储空间，默认含有20 个数组。

IMG_256

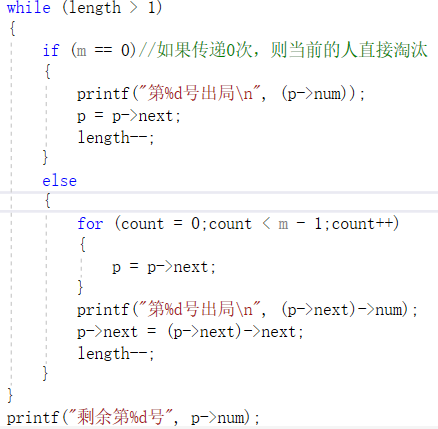
在这个数组中，第一个元素即Map[0]并不作为存储空间，而是作 为构造游标循环的一个媒介，初始元素从Map[1]开始。构造一个初 始化函数，元素中的next指针指向下一个元 素，并将最后一个 元素的next指针指向Map[1]，构造成一个循环游标。



之后，根据用户输入的人数值，再构建一个基于人数值的循环游标，为每一个结构体中加上人的序号：1、2、3···方法为根据人数创建相应数量的节点的指针P，利用Map[0]来为每一个P赋值。



构建完循环游标之后，只需进行最后一步：从Map[1]开始计数，每次计数后p = p->next，当计数到m后，便将当前的p清除，清除方式为将p的前一个节点与后一个节点连接起来。我们初始化两个int型常量length和count,分别存储当前游标的总长度和计数。每次找出一个人之后，length长度减1。当length = 1的时候，跳出循环，查找结束。



至此，便完成了约瑟夫环的游标法实现。

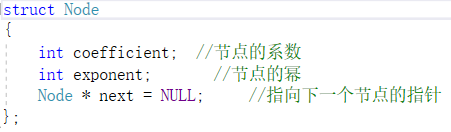
**多项式乘法：**

**指数不排序：**

分析：首先先定义好链表的结构。假设现在有p1、p2两个链表多项式，对于指数不排序的情况，可以采用直接比较的方法，将p1和p2的每个单项分别相乘，并将结果存放在一个新的指针链表中。在存放的过程中新的结果依次与链表中的各个单项相比较，保证链表中的元素是按幂从大到小排列的。

求解过程：

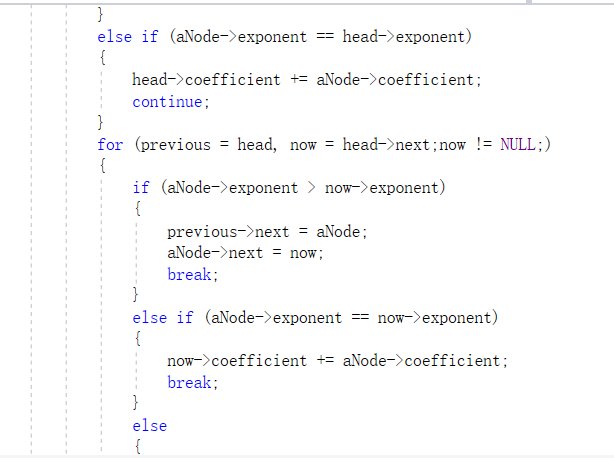
首先定义好链表的结构体，结构体内含有单项的系数和幂，以及指向下一个元素的结构体指针。

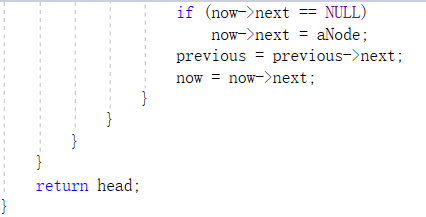


运用双重for循环，分别从p1和p2的第一个元素开始相乘。每次循环都会新建一个元素结构体用来存储相乘的结果。用head代表新的链表的头结点，然后将新建的节点依次与链表中的元素比较。分三种情况讨论：若新节点幂大于头结点的幂，则新节点变为头结点，head指向新节点；若新节点幂等于头结点或之后某个节点，则直接将新节点的系数与该节点相加；若新节点幂小于比较的节点，则指针往后移，重新比较，直到找到一个中间位置或者放到最后一个位置。

当双重for循环完成后，结果也已经存储到了新的链表中，直接返回新的链表即可。







时间复杂度分析：

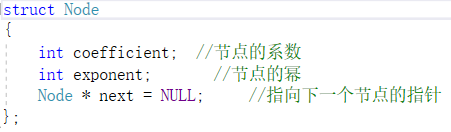
运用了双重for循环，假设p1和p2全部都是包含n个节点的链表，则一共会循环n^2次，进行n^2次乘法。在最坏的情况下，每一次的结果都要跟新链表中的所有元素进行比较才能成功插入，因此T(n) = n^2 + 1 + 2 + 3 +····+ n^2 = n^2(n^2+1)/2 + n^2 = O(n^4)。

**指数排序：**

分析：首先先定义好链表的结构。假设现在有p1、p2两个链表多项式，对于指数排序的情况，可以采用下列方法：因为p1、p2已经按幂的降序排列，取p1的任意一个元素与p2的所有元素顺序相乘构成的新链表也是一个降序排列。则问题可以变为将p1的每个元素依次与p2的所有元素顺序相乘构成的每个新链表相加。

求解过程：

首先定义好链表的结构体，结构体内含有单项的系数和幂，以及指向下一个元素的结构体指针。



然后定义链表相加函数。思路如下：

以最高次幂较大的一个链表为开头，将链表的下一个元素的幂与另一个链表的第一个元素的幂进行比较。

比较

6

2

3

4

5

7

根据比较结果选择较大的一方，将其与第一个节点连接起来，然后该节点跳到它的结构体中的next节点，重新进行比较连接。

比较

5

2

4

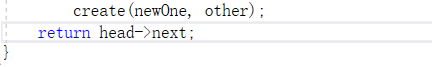
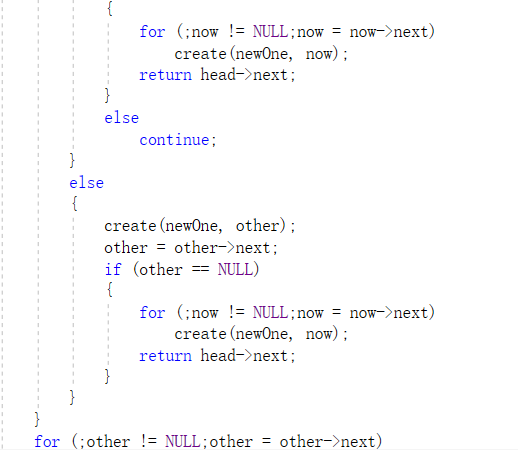
6

3

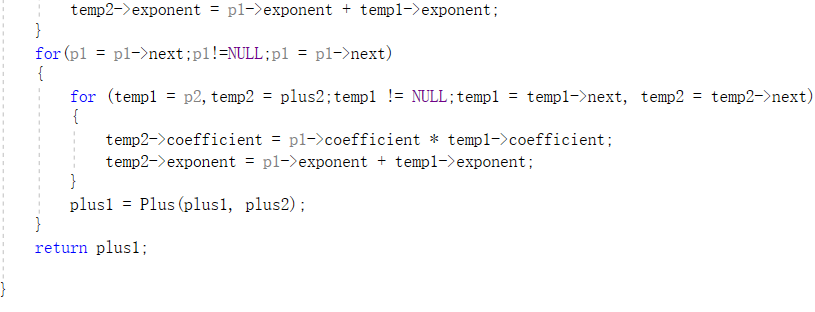
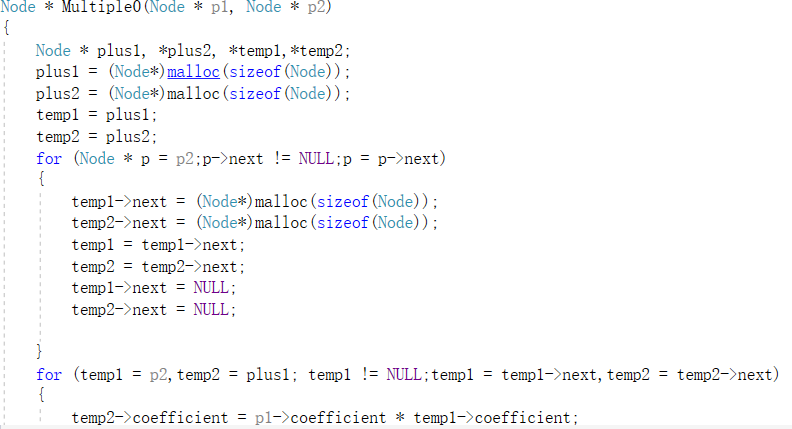
7

就这样一直进行比较与连接，直到全部比较连接完后形成了一条新的链表代表两条链表的和。





定义完相加函数之后，再定义乘法函数，在里面调用相加函数即可。利用for循环对p1进行遍历，将p1的每一个元素与p2的所有元素进行相乘构成一个新链表，然后每次遍历都调用相加函数把新链表加到已有的链表上去。最后返回链表即可。



时间复杂度分析：

假设p1、p2都是有n个节点的链表，对于相加函数，最坏的情况是进行2n次比较。在相乘函数里，调用for循环对p1进行遍历，并将p1的元素与p2所有元素相乘。所以一共有n^2次乘法与n-1次相加函数的调用。所以T(n) = n^2 + (n-1)\*2n = O(n^2)。