实验一 线性表（一）

按照教科书的指示，定义顺序表的结构，实现所有的顺序表运算函数：CreateList、InitList、DestroyList、IsEmpty、ListLength、DispList、GetElem、LocateElem、ListInsert和ListDelete，并在main函数中实现如下要求：

（1）定义一个类型为ElemType的数组，在里面存储任意10个元素，通过调用CreateList创建顺序表。

（2）调用GetElem函数，获取其中第n个元素的值，n由自己指定。

（3）调用两次LocateElem函数，第一次指定任意一个存在于顺序表中的元素，查看函数返回结果；第二次指定一个不存在于顺序表中的元素，查看函数返回结果。

（4）调用ListInsert和ListDelete函数任意次数，向顺序表中插入或删除任意元素。

（5）调用DispList显示顺序表的最终元素。

（6）程序结束前调用DestroyList释放顺序表空间。

实验二 线性表（二）

以下题目三选一：

（1）已知两个链表A和B分别表示两个集合，其元素递增排列。编一函数，求A与B的交集，并存放于A链表中。

（2）已知头指针分别为la和lb 的带头结点的单链表中，结点按元素值非递减有序排列。写出将la 和 lb两链表归并成一个结点按元素值非递减有序排列的单链表（其头指针为 lc）。

（3）已知p指向双向循环链表中的一个结点，其结点结构为data、next、prior三个域，写出算法change(p),交换p所指向的结点和它的前缀结点的顺序。

实验三 栈

以下题目三选一（本次实验允许使用C++的stl::Stack容器类及其它模板类）：

（1）从控制台输入两行整数，分别存放到两个数组中。第一行整数表示进栈顺序，第二行整数表示出栈顺序，要求判断出栈顺序是否合法。例如：

输入：

1 2 3 4 5

3 2 5 4 1

输出：

合法

（提示：假设两行整数分别存储在数组a和b中，构造一个栈，然后开始循环。每循环一次，数组a进栈一个元素，同时使用内层循环拿数组b中的元素与栈顶元素进行比较：如果b中当前下标对应的元素与栈顶元素一致，则出栈，下标加1，继续下一轮循环，否则就退出循环。如果所有循环结束，栈还不是空的，则出栈顺序不合法。**本提示请不要直接写到实验报告中，最好进行适当组织后分散出现在实验报告的相应位置**）

（2）愚人节礼物

四月一日那天，玛丽亚想到了一个整人的方法——给好友送“礼物”。“礼物”由一堆盒子组成，每个盒子里面又可以套一个或多个盒子，只有一个盒子里面装着真正的礼物，并且该盒子中不会再放别的盒子。

现在用成对的括号()表示盒子，字母B表示礼物，因此(B)就表示装有礼物的盒子了。玛利亚想要你帮她计算出愚人指数，即**最少**拆多少个盒子才能拿到礼物。

在控制台输入一组字符串，由(、)和B这三种字符组成，输出一个整数，表示愚人指数。例如：

输入 (B) 输出 1

输入 ((()((B)())())) 输出 4

如果括号不匹配，输出“不匹配”

（3）表达式运算：

已知表达式由数字和运算符组成，运算符为：+、-、\*、/、(、)和^，其中向上箭头表示乘方运算，如2^3表示2的3次方，其优先级要高于乘法和除法。

基本要求：运算符两侧的数字只为个位数，只需求把中缀表达式转换成后缀表达式即可。例如：

输入 3+2\*5^4+6\*3 输出：3254^\*+63\*+

高级要求：

（1）运算符两侧的数字可以是两位数，此时的后缀表达式中需要用#把各数字隔开。例如：

输入 (32+30)\*10^2 输出：32#30#+10#2#^\*

（2）求出表达式中的值，即输入一个表达式，直接输出它的值，不需要输出后缀表达式。

（注：两个高级要求可以任选一个，也可以都选，需要在实验报告里说明本次实验完成了哪些要求，例如“本实验只实现了基本要求”或“本实验实现了高级要求（2）”之类的语句，作为3.2节的“数据结构定义”第一段。**本段注释说明请不要写到实验报告中**）。

实验四 队列

以下题目三选一（本次实验允许使用C++自带的模板类）：

（1）约瑟夫环

一共有n个人围成一个圆圈，每个人的编号分别1、2、…、n。第1个人从1开始顺时针报数，报到m的人令其出列。然后再从下一个人开始，从1顺时针报数，报到第m个人，再令其出列。……如此下去，直到圆圈中只剩下一个人为止，此人即为优胜者。

现在要求输入n和m，输出优胜者的原始编号。

例如输入 8 3 输出 7

（2）舞伴问题

在一场舞会中有m位男生和n位女生，分别站成两队。每当一首舞曲响起，队首的男女生配对进入舞池开始跳舞；舞曲结束后，这两人便分别进入各自的队尾。现在要求在控制台输入男生的数目（m）、女生的数目（n）、舞曲的数目（t），然后输入每位男生和每位女生的姓名，最后输出舞伴的配对情况，例如：

曲1：张明 ---- 徐艳

曲2：李兵 ---- 杨丽

…

**要求输入的男女生数目不能相等，并且舞曲的数目要大于男女生数目的最大值**。

（3）打牌游戏

将一副牌平均分为两份，每人拿一份。A先拿出手中的第一张扑克牌放在桌上，然后B也拿出手中的第一张扑克牌，并放在A刚打出的扑克牌的上面，就像这样两人交替出牌。出牌时如果某人打出的牌与桌上某张牌的牌面相同时，即可将两张相同的牌及其中间所夹的牌全部取走，并依次放到自己手中牌的末尾，当其中一人手中的牌全部出完时，游戏结束，对方获胜。

规定：牌面由1~n的正整数组成，只包含两张，每副牌的数量是2n。首先输入n，然后再输入两行正整数，分别表示A和B手中的牌。输出获胜者、它手中的牌以及桌上的牌。例如：

输入：

6

2 4 1 2 5 6

3 1 3 5 6 4

输出：

获胜者为A

A手中的牌：1 6 5 2 3 4 1

桌上的牌：2 6 5 4 3

（提示：每个人手上的牌可以用（环形）队列实现，打牌即为出队操作，收牌即为入队操作。桌子就是一个栈，每打一张牌即进栈，取牌即出栈。另外使用一个大小为n+1的临时数组，各元素初值为0，程序运行过程中，如果数组[5]=1，表示栈内（桌上）有一张牌的牌面为5，用于判断是往桌上放牌（压栈）还是从桌上取牌（批量出栈），如果是取牌请不要忘记将数据下标对应的元素值重置为0。**该提示请不要写到实验报告中**。）

实验五 字符串

以下题目三选一：

（1）字符串压缩

若某个字符串中存在大量、连续的字符，则可以进行压缩存储，方法为：先存储连续出现的字符数目，接着存储该字符，如果是单个字符，则只存储该字符本身。例如：

输入的字符串为：aabbbbcbbbddddddaa

则输出：2a4bc3b5d2a

（2）字符串交叉

输入两个字符串，要求对它们进行交叉，并把结果保存到一个新字符串中。（**如果只是输出交叉结果，将会酌情扣分，括号里的这句话请不要写到实验报告中**）例如：

输入两个字符串：

abcd

1234

输出：

a1b2c3d4

如果两个字符串长度不同，则先交叉，然后把剩下的追加到后面，例如：

输入两个字符串：

abc

123

输出：

a1b2c3def

（3）单词逆序

在控制台输入一句英文，要求句中的每个单词逆序输出。

例如输入：how old are you 输出：woh dlo era uoy

提示：可使用队列存放每个单词，栈存放单词中的每个字母。

实验六 二叉树（一）

以括号表示法作为输入字符串构造一棵二叉树，完成下列实验要求：

（1）如果输入字符串不符合规范，显示错误信息：“输入错误”。

（2）分别显示该二叉树的先序、中序和后序遍历。

（3）求出该二叉树的高度。

（4）输出该二叉树所有的叶子结点。

（5）找出二叉树中的最大、最小值结点，**并写出它们所在的层数（此项可选）**。

实验七 二叉树（二）

以下题目二选一：

（1）根据先序遍历和中序遍历字符串构造二叉树，并分别以括号表示法输出结果，如果输入的字符串不符合规范，需要显示错误信息：“输入错误”或者“无法构建二叉树”。例如：

输入：

ABDGCEF

DGBAECF

输出：

A(B(D(,G)),C(E,F))

（2）给定英文字母的种类以及权重，构造哈夫曼树，并输出每个字母的哈夫曼编码，以及哈夫曼树的带权路径长度WPL。

输入：

8

a b c d e f g h

0.07 0.19 0.02 0.06 0.32 0.03 0.21 0.1

其中第一行输入不同英文字母的数目，第二行输入各个英文字母，第三行输入每个英文字母对应的权重。

输出：

a 1010

b 00

c 10000

d 1001

e 11

f 10001

g 01

h 1011

WPL=2.61

实验八 图

以下题目四选一，注意要在“数据结构的定义”中注明使用邻接矩阵还是邻接表，并有相关的代码定义。

（1）

设计算法以实现对无向图G的深度优先遍历，要求：将每一个连通分量中的顶点以多行的形式输出。例如，下图的输出结果为：

1 3

2 6 7 4 5 8

9 10

9

5

4

2

1

10

8

7

6

3

（2）

编写算法求下图中任意两个顶点的简单路径，每条路径显示为一行。

例如：

输入：

5 8

输出：

5 1 2 3 6 8

5 1 4 7 8

5 6 8

（3）

编写算法求下图的最小生成树，要求打印出该生成树的每条边和对应权值。

3

2

6

4

9

5

8

7

10

（4）编写算法打印出下图的一个拓扑序列（如能打印所有拓扑序列可以加分）。