**作业 HW1 实验报告**

姓名：陆明奇 学号：2050283 日期：2021年10月15日

1. **涉及数据结构和相关背景**

本实验所涉及的数据结构为线性表，它是线性结构的一种。线性结构的特点是：在数据元素的非空有限集中，（1）存在唯一的一个被称作“第一个”的数据元素；（2）存在唯一的一个被称作“最后一个”的数据元素；（3）除第一个之外，集合中的每个数据元素均只有一个前驱；（4）除最后一个之外，集合中每个元素均只有一个后驱。线性表是n个数据元素的有限序列，是一个相当灵活的数据结构，它的长度可根据需要增长或缩短。对于线性表中的数据元素，不仅可以进行访问，还可以进行插入和删除等。

**2. 实验内容**

**2.1 学生信息管理**

**2.1.1 问题描述**

定义一个包含学生信息（学号，姓名）的顺序表，使其具有如下功能：(1) 根据指定学生个数，逐个输入学生信息；(2) 给定一个学生信息，插入到表中指定的位置；(3) 删除指定位置的学生记录；(4) 分别根据姓名和学号进行查找，返回此学生的信息；(5) 统计表中学生个数。

**2.1.2 基本要求**

insert i 学号 姓名: 在第i个位置插入学生信息, 若i位置不合法，输出-1，否则输出0

remove j:表示删除第j个元素，若元素位置不合适，输出-1，否则输出0

check name 姓名y：查找姓名y在顺序表中是否存在，若存在，输出其位置序号及学号、姓名，若不存在，输出-1

check no 学号x：查找学号x在顺序表中是否存在，若存在，输出其位置序号及学号、姓名，若不存在，输出-1。

**2.1.3 数据结构设计**

本题采用了顺序表。顺序表是采用顺序存储结构的线性表，它利用内存中的一片连续存储区域存放学生信息。可以根据需要对表中的所有学生信息进行访问，学生信息的插入和删除可以在表中的任何位置进行。

**2.1.4功能说明（函数、类）**

bool ListInsert(student\* &stu, int pos, char no1[8], char name1[20], int &n)

功能：在表中某一位置插入一个学生的信息

Para stu: 学生信息表头地址；

Para pos：插入位置；

Para no1[18]：学生学号；

Para name1[20]：学生姓名；

Para n：学生数量；

返回值：是否插入成功；

bool ListDelete(student\* &stu, int id, int &n)

功能：在表中某一位置删除一个学生的信息

Para stu: 学生信息表头地址；

Para id：删除学生的位置；

Para n：学生数量；

返回值：是否删除成功；

int ListSearchName(student\* &stu, char name1[20], int &n)

功能：在表中通过姓名寻找一个学生的信息

Para stu: 学生信息表头地址；

Para name1[20]：学生姓名；

Para n：学生数量；

返回值：该学生在表中的位置；

int ListSearchNo(student\* &stu, char no1[8], int &n)

功能：在表中通过学号寻找一个学生的信息

Para stu: 学生信息表头地址；

Para no1[18]：学生学号；

Para n：学生数量；

返回值：该学生在表中的位置；

**2.1.5 调试分析（遇到的问题和解决方法）**

由于学生数量的不断变化，我通过动态申请的方式来获取学生信息的存储空间。每当有学生信息删除时，我一开始是用free()函数来释放内存，但运行时出现了弹窗错。经过思考，我采用了realloc（）的方式重新申请n-1个地址空间，变相地达到了释放被删除学生内存的操作。其次，一开始做题时没有看清题目，学生学号定义为了int型变量，好多个测试点没有过。通过仔细阅读题目，发现学生学号和姓名都需要用字符串变量来定义。

**2.1.6 总结和体会**

通过这道题，我对于顺序表有了更深入的了解。能熟练实现顺序表的创建，第i个位置插入一个新的元素、删除第i个元素、查找某元素、顺序表的销毁等操作。我认为这道题最大的易错点在于变量类型的定义，一般学号都定义为int型，但这道题将其规定为字符串型，如果输出int型的学号无法通过测试点。其次，动态申请内存空间以及释放也需要当心，不能多次释放同一个空间，也不能释放不是动态申请的空间。

**2.2 有序表的合并**

**2.2.1 问题描述**

已知线性表LA和LB中的元素按值非递减有序排列，现将LA和LB归并成一个新的有序表LC，且LC中的数据元素仍按值非递减有序排列。

**2.2.2 基本要求**

依次输入该两个有序表，输入时可以是非有序的，输出合并后的结果。

**2.2.3 数据结构设计**

这道题目中两个表中的数据元素不再发生插入删除操作，因此我采用顺序表。设置两个指针分别指向LA和LB中某个元素，进行合并操作。算法复杂度为O（La.length+Lb.length）。

**2.2.4功能说明（函数、类）**

void MergeList(int\* la, int\* lb, int\* &lc ,int lena,int lenb)

功能：将表LA和LB合并为LC；

Para la：表LA首元素的地址；

Para lb：表LB首元素的地址；

Para lc：表LC首元素的地址；

Para lena：表LA的长度；

Para lenb：表LB的长度；

返回值：无；

**2.2.5 调试分析（遇到的问题和解决方法）**

当指针指向LA表或LB表末尾后，直接结束了赋值操作，没有将长度较长的表中的剩余元素进行插入，导致错误。应该在退出循环后再次判断是否两个指针都指向了表尾。

**2.2.6 总结和体会**

这道题使我掌握了顺序表合并的实现过程。在实现中，需要注意只有当两个表的指针都指向了表尾时，合并才算结束。

**2.3 扑克牌游戏**

**2.3.1 问题描述**

扑克牌有4种花色：黑桃、红心、梅花、方块。每种花色有13张牌，编号从小到大为：A,2,3,4,5,6,7,8,9,10,J,Q,K。对于一个扑克牌堆，定义以下4种操作命令：1) Append：添加一张扑克牌到牌堆的底部。2) Extract:从牌堆中抽取某种花色的所有牌,按照编号从小到大进行排序,并放到牌堆的顶部。3)Revert：使整个牌堆逆序。4)Pop：如果牌堆非空，则除去牌堆顶部的第一张牌，并打印该牌的花色和数字；如果牌堆为空，则打印NULL。

**2.3.2 基本要求**

每次输入一个命令，当收到Pop指令后输出一行（花色和数子或NULL），最后将牌堆中的牌从牌堆顶到牌堆底逐一输出（花色和数字），若牌堆为空则输出NULL。

**2.3.3 数据结构设计**

这道题需要对扑克牌进行插入，删除，逆置，抽取等操作，若采取链表将十分繁琐，我选择采用vector来完成这道题。Vector可以动态地改变数组的大小，还内置了逆置函数，能方便地进行插入，删除，逆置，抽取等操作。

**2.3.4功能说明（函数、类）**

bool cmp(poker v1, poker v2)

功能：自定义sort函数的比较方式；

Para v1：比较对象；

Para v2：比较对象：

返回值：比较的结果；

void read（）

功能：读取输入的指令，花色以及数字；

参数：无：

返回值：无；

**2.3.5 调试分析（遇到的问题和解决方法）**

在利用迭代器遍历查找数组中要抽取的元素时，我一开始的做法是：找到特定元素并使用erase（）函数删去该元素。但这种做法在运行时异常返回了。通过查阅资料，我了解到使用erase函数删掉一个元素后，其后的元素都会向前移动，所以此时指向下一个元素的迭代器其实跟刚刚被删除元素的迭代器是一样的。这样会导致迭代器失效。想到erase（）函数的返回值是指向下一个元素的迭代器，因此可以将erase（）函数的返回值赋值给遍历查找时的迭代器it。

**2.3.6 总结和体会**

这道题使我了解到了vector的一些实用技巧以及相关内置函数。本题需要注意到当牌堆中没有牌时，应该输出NULL，而不是什么都不输出。

**2.4 一元多项式的相加和相乘**

**2.4.1 问题描述**

一元多项式是有序线性表的典型应用，用一个长度为m且每个元素有两个数据项（系数项和指数项）的线性表((p1,e1),(p2,e2),...,(pm,em))可以唯一地表示一个多项式。本题需要实现多项式的相加和相乘运算。

**2.4.2 基本要求**

依次输入两个多项式每项的系数和指数（不保证有序）。第5行输入一个整数，若为0,执行加法运算并输出结果，若为1，执行乘法运算并输出结果；若为2，输出一行加法结果和一行乘法的结果。输出要求指数从小到大。

**2.4.3 数据结构设计**

考虑到多项式的次数可能很高且变化很大，我采用链表这一数据结构。一个链表存储一个多项式的信息。每次相加相乘得到的新的一项也可以快速地插入到原链表之中。

**2.4.4功能说明（函数、类）**

node\* creat\_list(int n)

功能：读入链表并排序

Para n：生成链表的场地；

返回值：链表的头指针；

node\* add\_list(node\* a, node\* b)

功能：将两个多项式进行求和；

Para a：多项式一的头指针；

Para b：多项式二的头指针；

返回值：和多项式的头指针；

node\* multi\_list(node\* a, node\* b)

功能：将两个多项式进行求积；

Para a：多项式一的头指针；

Para b：多项式二的头指针；

返回值：积多项式的头指针；

void print\_list(node\* l)

功能：将多项式进行输出；

Para l：多项式的头指针；

返回值：无；

**2.4.5 调试分析（遇到的问题和解决方法）**

一开始相乘和相加是在原多项式上进行插入和删除项的操作，操作完之后常常会忘记将指针归位，导致错误。之后我为了防止修改指针本身的值，使用代理指针来完成操作，也就是游标。

**2.4.6 总结和体会**

这道题使我掌握了链表的一些基本操作，包括插入，删除等。本题需要注意的是两个多项式对应系数相加，若其和不为0，则构成新多项式中的一项。若为0，则应该把这一项给删除。

**2.5 级数相加**

**2.5.1 问题描述**

在一行中输出级数A+2A2+3A3+。。。+NAN =的整数值。

**2.5.2 基本要求**

每一行中给出整数N和A的值，（1<=N<=150，0<=A<=15），输出答案。

**2.5.3 数据结构设计**

由于答案可能超过int的范围，我选择采用字符串string来模拟每一位的相加和相乘。

**2.5.4功能说明（函数、类）**

string add(string str1, string str2)

功能：高精度加法；

Para str1：加数1；

Para str2：加数2；

返回值：两数之和；

string mul(string str1, string str2)

功能：高精度乘法；

Para str1：乘数1；

Para str2：乘数2；

返回值：两数之积；

**2.5.5 调试分析（遇到的问题和解决方法）**

在模拟加法时，没有提前判断两个数的位数而直接相加，导致位数多的那个加数没有数可加。因为两个数的位数不一定相等，需要在相加前判断两个加数的位数，在位数少的那个数的前面补0，使两个数位数相等。

**2.5.6 总结和体会**

这道题使我了解到了大数加法和乘法的实现方式-用数组模拟每一位的相加。在相加时，需要特别注意两个数的位数是否相同，不同的话需要在位数少的那个数前面补0。

**3.实验总结**

这次的实验主题是线性表，线性表中数据元素之间的关系是一对一的关系，即除了第一个和最后一个数据元素之外，其它数据元素都是首尾相接的。但也有例外，比如，[循环链表](https://baike.baidu.com/item/%E5%BE%AA%E7%8E%AF%E9%93%BE%E8%A1%A8/3228465)逻辑层次上也是一种线性表，它存储层次上属于链式存储，但是把最后一个数据元素的尾指针指向了首位结点。线性表的[逻辑结构](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E7%BB%93%E6%9E%84/9663235)简单，便于实现和操作。因此，线性表这种数据结构在实际应用中是广泛采用的一种数据结构。