**郑 州 轻 工 业 学 院**

**实 验 报 告**

**课程名称：** 数据结构

**实验名称：** 一元多项式的计算

**院 （系）：** 计算机与通信工程学院

**姓 名：** 刘亚炜

**学 号：** 541707090128

**专业班级：** 物联网工程17-1

**指导教师：** 孟颍辉

2019 **年** 6 **月** 5 **日**

**实验报告成绩评定表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评定项目 | 内 容 | 满 分 | 评 分 | 总 分 |
| 实验态度 | 实验认真，态度端正，遵守纪律，出勤情况。 | 20 |  |  |
| 实验过程 | 按实验要求完成各种功能或操作，代码书写规范，注释清晰，设计严谨，运行结果正确。 | 40 |  |
| 报告撰写 | 报告字迹整洁、内容丰富、条理清楚；图、表、文字表达准确规范，上交及时。 | 40 |  |
| 总成绩 | 采用五级分制：优、良、中、及格、不及格 |  | | |
| **评语：**  **指导老师签字： 年 月 日** | | | | |

实验报告正文

1. **实验目的**

掌握顺序表和单链表的存储特点及插入、删除等算法。

1. **实验内容及要求**
2. 任务描述

灵活运用顺序表和单链表的相关算法实现一元多项式的计算。

1. 主要数据类型与变量

（**必要时，可对数据类型和变量进一步解释或说明，增加可读性**）

bool flag = 0;//表创建的标志

typedef struct ElemType{

int a, b;

ElemType(int aa, int bb){

this->a = aa;

this->b = bb;

}

};

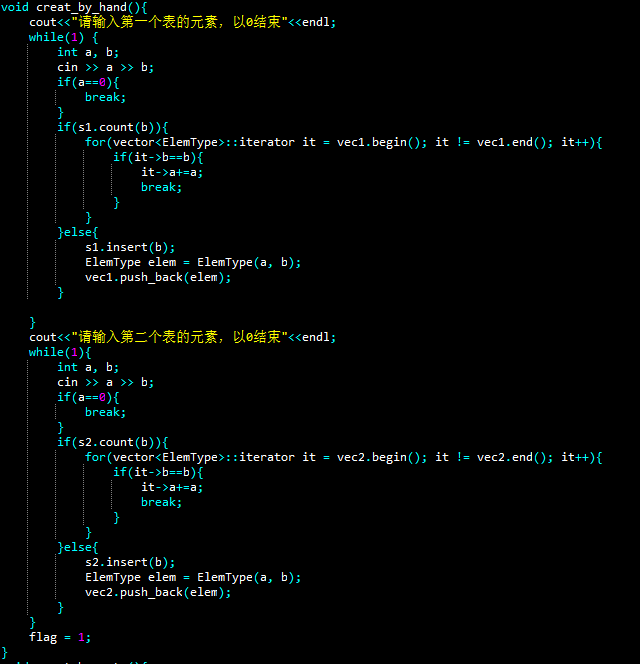
vector<ElemType> vec1;

vector<ElemType> vec2;

set<int> s1;

set<int> s2;

1. 算法或程序模块



1. **测试**
   1. 方案

程序中附加有自动生成数据的操作并且无规律，可充分保证程序的稳定性。

* 1. 结果



1. **总结与讨论**

本实验采取的是快速排序，并且保证了数据的唯一性和稳定性，并且使用了动态数组，经过多次测试，保证了该程序在10万以上的数据秒出。

**附：程序的源代码**

**#include <bits/stdc++.h>**

**#include <vector>**

**#include <set>**

**using namespace std;**

**bool flag = 0;//表创建的标志**

**typedef struct ElemType{**

**int a, b;**

**ElemType(int aa, int bb){**

**this->a = aa;**

**this->b = bb;**

**}**

**};**

**vector<ElemType> vec1;**

**vector<ElemType> vec2;**

**set<int> s1;**

**set<int> s2;**

**bool cmp1(ElemType e1, ElemType e2){//从小到大排序**

**if(e1.b!=e2.b){**

**return e1.b<e2.b;**

**}**

**return e1.a<e2.a;**

**}**

**bool cmp2(ElemType e2, ElemType e1){//从大到小排序**

**if(e1.b!=e2.b){**

**return e1.b<e2.b;**

**}**

**return e1.a<e2.a;**

**}**

**void menu(){**

**cout<<"-----一元多项式计算器-----"<<endl;**

**cout<<"1.手动创建两个一元多项式"<<endl;**

**cout<<"2.自动创建两个一元多项式"<<endl;**

**cout<<"3.添加一个元素/插入一个元素"<<endl;**

**cout<<"4.删除元素"<<endl;**

**cout<<"5.使两个表有序"<<endl;**

**cout<<"6.直接输出两个表(遍历)"<<endl;**

**cout<<"7.清除两个表的所有数据"<<endl;**

**cout<<"0.退出该系统"<<endl;**

**}**

**void creat\_by\_hand(){**

**cout<<"请输入第一个表的元素，以0结束"<<endl;**

**while(1) {**

**int a, b;**

**cin >> a >> b;**

**if(a==0){**

**break;**

**}**

**if(s1.count(b)){**

**for(vector<ElemType>::iterator it = vec1.begin(); it != vec1.end(); it++){**

**if(it->b==b){**

**it->a+=a;**

**break;**

**}**

**}**

**}else{**

**s1.insert(b);**

**ElemType elem = ElemType(a, b);**

**vec1.push\_back(elem);**

**}**

**}**

**cout<<"请输入第二个表的元素，以0结束"<<endl;**

**while(1){**

**int a, b;**

**cin >> a >> b;**

**if(a==0){**

**break;**

**}**

**if(s2.count(b)){**

**for(vector<ElemType>::iterator it = vec2.begin(); it != vec2.end(); it++){**

**if(it->b==b){**

**it->a+=a;**

**break;**

**}**

**}**

**}else{**

**s2.insert(b);**

**ElemType elem = ElemType(a, b);**

**vec2.push\_back(elem);**

**}**

**}**

**flag = 1;**

**}**

**void creat\_by\_auto(){**

**cout<<"请输入你要创建的表一长度"<<endl;**

**int length1 = 0;**

**cin >> length1;**

**srand((int)time(0));**

**for(int i = 0; i < length1; i++) {**

**int a = rand()%100;**

**int b = rand()%100;**

**if(s1.count(b)){**

**for(vector<ElemType>::iterator it = vec1.begin(); it != vec1.end(); it++){**

**if(it->b==b){**

**it->a+=a;**

**break;**

**}**

**}**

**}else{**

**s1.insert(b);**

**ElemType elem = ElemType(a, b);**

**vec1.push\_back(elem);**

**}**

**}**

**cout<<"请输入你要创建的表二长度"<<endl;**

**int length2 = 0;**

**cin >> length2;**

**for(int i = 0; i < length2; i++) {**

**int a = rand()%100;**

**int b = rand()%100;**

**if(s2.count(b)){**

**for(vector<ElemType>::iterator it = vec2.begin(); it != vec2.end(); it++){**

**if(it->b==b){**

**it->a+=a;**

**break;**

**}**

**}**

**}else{**

**s2.insert(b);**

**ElemType elem = ElemType(a, b);**

**vec2.push\_back(elem);**

**}**

**}**

**flag = 1;**

**}**

**void creat\_elem(){**

**cout<<"请输入你想创建的元素数量"<<endl;**

**int length = 0;**

**cin >> length;**

**for(int i = 0; i < length; i++) {**

**cout<<"请输入a, b"<<endl;**

**int a, b;**

**cin >> a >> b;**

**cout<<"你想要插入表一还是表二？"<<endl;**

**int c;**

**cin >> c;**

**ElemType elem = ElemType(a, b);**

**if(c==1){**

**vec1.push\_back(elem);**

**}else if(c==2){**

**vec2.push\_back(elem);**

**}**

**cout<<"插入成功"<<endl;**

**}**

**}**

**void delete\_elem(){**

**cout<<"请输入你想删除元素的方式"<<endl;**

**cout<<"1.删除最末尾元素"<<endl;**

**cout<<"2.删除满足条件的元素"<<endl;**

**cout<<"3.删除一定范围内的元素"<<endl;**

**cout<<"4.精确删除某个元素"<<endl;**

**cout<<"5.清空所有元素"<<endl;**

**cout<<"6.取消删除"<<endl;**

**int ans;**

**vector<ElemType>::iterator it;**

**cin >> ans;**

**switch(ans){**

**case 1:**

**cout<<"你想删除哪个表的最末尾元素？"<<endl;**

**int ans2;**

**cin >> ans2;**

**if(ans2==1){**

**vec1.pop\_back();**

**}else if(ans2==2){**

**vec2.pop\_back();**

**}**

**break;**

**case 2:**

**cout<<"请输入你想要删除系数为几的元素"<<endl;**

**int an2;**

**cin >> an2;**

**cout<<"你想删除表一(输入1)，表二(输入2)，还是都删(输入3)？"<<endl;**

**int ans3;**

**cin >> ans3;**

**if(ans3==1){**

**for(it == vec1.begin(); it != vec1.end(); it++) {**

**if(it->b==an2){**

**vec1.erase(it);**

**it--;**

**}**

**}**

**}else if(ans3==2){**

**for(it == vec2.begin(); it != vec2.end(); it++) {**

**if(it->b==an2){**

**vec2.erase(it);**

**it--;**

**}**

**}**

**}else if(ans3==3){**

**for(it == vec1.begin(); it != vec1.end(); it++) {**

**if(it->b==an2){**

**vec1.erase(it);**

**it--;**

**}**

**}**

**for(it == vec2.begin(); it != vec2.end(); it++) {**

**if(it->b==an2){**

**vec2.erase(it);**

**it--;**

**}**

**}**

**}**

**break;**

**case 3:**

**cout<<"请输入你想要删除的表及其范围"<<endl;**

**cout<<"你想删除表几的元素？"<<endl;**

**int as2;**

**cin >> as2;**

**if(as2==1){**

**cout<<"表一可以删除的范围为0到"<<vec1.size()<<endl;**

**cout<<"你想删除的范围？" <<endl;**

**int l, r;**

**cin >> l >> r;**

**for(it = l + vec1.begin(); it != r + vec1.begin(); it++){**

**vec1.erase(it);**

**it--;**

**}**

**}else if(as2==2){**

**cout<<"表二可以删除的范围为0到"<<vec2.size()<<endl;**

**cout<<"你想删除的范围？" <<endl;**

**int l, r;**

**cin >> l >> r;**

**for(it = l + vec2.begin(); it != r + vec2.begin(); it++){**

**vec2.erase(it);**

**it--;**

**}**

**}**

**break;**

**case 4:**

**cout<<"你想删除哪个表中的元素？"<<endl;**

**int ns2;**

**cin >> ns2;**

**if(ns2==1){**

**cout<<"表一可以删除的范围为0到"<<vec1.size()<<endl;**

**cout<<"请输入你想要删除的元素的下标"<<endl;**

**int ans3;**

**cin >>ans3;**

**it = vec1.begin() + ans3;**

**vec1.erase(it);**

**}else if(ns2==2){**

**cout<<"表二可以删除的范围为0到"<<vec2.size()<<endl;**

**cout<<"请输入你想要删除的元素的下标"<<endl;**

**int ns3;**

**cin >>ns3;**

**it = vec2.begin() + ns3;**

**vec1.erase(it);**

**}**

**break;**

**case 5:**

**vec1.clear();**

**vec2.clear();**

**flag = 0;**

**case 6:**

**return;**

**}**

**}**

**void sort\_elem(){**

**cout<<"你想从大到小还是从小到大？"<<endl;**

**cout<<"1.从小到大"<<endl;**

**cout<<"2.从大到小"<<endl;**

**int ans;**

**cin >> ans;**

**if(ans==1){**

**sort(vec1.begin(), vec1.end(), cmp1);**

**sort(vec2.begin(), vec2.end(), cmp1);**

**}else if(ans==2){**

**sort(vec1.begin(), vec1.end(), cmp2);**

**sort(vec2.begin(), vec2.end(), cmp2);**

**}**

**cout<<"排序完成"<<endl;**

**}**

**void syso(){**

**cout<<"表一：Am(X)="<<endl;**

**for(vector<ElemType>::iterator it = vec1.begin(); it != vec1.end(); it++) {**

**if(it==vec1.begin()){**

**cout<<it->a<<"X^"<<it->b;**

**}else{**

**cout<<"+"<<it->a<<"X^"<<it->b;**

**}**

**}**

**cout<<endl;**

**cout<<"表二：Bn(X)="<<endl;**

**for(vector<ElemType>::iterator it = vec2.begin(); it != vec2.end(); it++) {**

**if(it==vec2.begin()){**

**cout<<it->a<<"X^"<<it->b;**

**}else{**

**cout<<"+"<<it->a<<"X^"<<it->b;**

**}**

**}**

**cout<<endl;**

**}**

**int main(){**

**while(1){**

**menu();**

**int ans;**

**cin >> ans;**

**switch(ans){**

**case 1:**

**creat\_by\_hand();**

**break;**

**case 2:**

**creat\_by\_auto();**

**break;**

**case 3:**

**if(flag){**

**creat\_elem();**

**}else{**

**cout<<"你还没创建表，请先创建表"<<endl;**

**}**

**break;**

**case 4:**

**if(flag){**

**delete\_elem();**

**}else{**

**cout<<"你还没创建表，请先创建表"<<endl;**

**}**

**case 5:**

**if(flag){**

**sort\_elem();**

**}else{**

**cout<<"你还没创建表，请先创建表"<<endl;**

**}**

**break;**

**case 6:**

**syso();**

**break;**

**case 7:**

**vec1.clear();**

**vec2.clear();**

**flag = 0;**

**break;**

**case 0:**

**return 0;**

**}**

**}**

**}**