山东大学 计算机科学与技术 学院

数据结构与算法 课程实验报告

学号：202200130048	姓名： 陈静雯	班级： 6
实验题目：数组描述线性表		
实验学时：2	实验日期： 10.12	

实验目的：
封装线性表类，提供插入，删除，查找等操作
线性表实现使用数组描述方法（顺序存储结构）

软件开发工具：
Vscode

1. 实验内容
设通讯录中每一个联系人的内容有：姓名、电话号码、班级、宿舍。由标准输入读入联系人信息，使用线性表中操作实现通讯录管理功能，包括：插入、删除、编辑、查找（按姓名查找）；键盘输入一班级，输出通讯录中该班级中所有人的信息。

2. 数据结构与算法描述 （整体思路描述，所需要的数据结构与算法）
用到线性表类，结合结构体，实现通讯录的功能
Insert：首先判断线性表容量是否已等于大小，若是先扩充容量成两倍，再放入线性表
Erase：遍历线性表找到该元素，后面的元素前移进行补位，最后 size--，若找不到，则无变化
Find：遍历线性表，找到输出 1，没有输出 0
Change：遍历线性表，找到要修改的元素进行修改。

3. 测试结果（测试输入，测试输出）

```
28
0 Evan 57298577609 1 65
0 WINNIE 37367348390 4 1
3 Evan
1
4 6
0
3 WINNIE
1
1 Evan
4 7
0
1 WINNIE
3 MARYAM
0
3 CAMERON
0
3 TZIVIA
0
0 OMAR 16447001130 6 55
4 8
0
4 2
0
3 JADEN
0
3 ELIZABETH
0
2 OMAR 1 79409905568
3 JOSHUA
0
2 OMAR 1 8978214817
1 OMAR
3 Azaan
0
3 MARIA
0
0 HANNAH 94060479192 5 98
3 HEIDY
0
1 HANNAH
0 Axel 92066832927 3 70
1 Axel
3 TIFFANY
0
```

4. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）

测试答案错误，原因是函数应该用引用传递，却用了值传递，导致变量值没有真正改变。

5. 附录：实现源代码（本实验的全部源程序代码，程序风格清晰易理解，有充分的注释）

```
#include <iostream>
using namespace std;

struct telephone{
    string name;        //姓名
    string tele;        //电话
    string clas;        //班级
    int domi;           //宿舍
    bool operator==(const telephone b) const{    //判等运算符重载
        if(this->name==b.name) return true;
        else return false;
    }
};
```

```
template <class T>
class myarray{
public:
    myarray(int capacity);
    void insert(T newelement);
    void erase(T thelement);
    void find(T thelement);
    void change(T thelement,int p,string t);
    void output(string t);
    int arraysize(){
        return size;
    }
private:
    int length;
    T* element;
    int size;
};

template <class T>
myarray<T>::myarray(int capacity){   //构造函数
    element = new T[capacity];
    length=capacity;
    size=0;
}

template <class T>
void myarray<T>::insert(T newelement){   //在数组最后插入新元素
    if(size==length){                    //若空间不够，重新进行动态分配
        length*=2;
        T* temp = new T [length];
        for(int i=0;i<size;i++){
            temp[i]=element[i];
        }
        element=temp;
    }
    element[size++]=newelement;     //队尾插入
}

template <class T>
void myarray<T>::erase(T thelement){     //删除
    int i=0;
    bool check = false;          //标记是否找到元素
    for(;i<size;i++){
        if(element[i]==thelement){
            check=true;
            break;   //找到要找的元素
```

```
        }
    }
    for(int j=i;j<size-1;j++){
        element[j]=element[j+1];        //删除该元素，后面的元素依次前移
    }
    if(check) size--;
}

template<class T>
void myarray<T>::find(T thelement){    //查找
    for(int i=0;i<size;i++){
        if(element[i]==thelement){   //找到输出 1
            cout<<1<<endl;
            return ;
        }
    }
    cout<<0<<endl;
}

void change_address(telephone &temp,int p,string t){   //结构体的修改
    if(p==1) temp.tele=t;
    if(p==2) temp.clas=t;
    if(p==3) temp.domi=atoi(t.c_str());
}

template <class T>
void myarray<T>::change(T thelement,int p,string t){   //对对应元素进行修改
    for(int i=0;i<size;i++){
        if(element[i]==thelement){
            change_address(element[i],p,t);
            break;
        }
    }
}

void sum_domi(telephone* temp,int leng,string t){   //计算异或和
    int sum=0;
    for(int i=0;i<leng;i++){
        if(temp[i].clas==t) sum^=temp[i].domi;
    }
    cout<<sum<<endl;
}

template <class T>
void myarray<T>::output(string t){
    sum_domi(element,this->arraysize(),t);
```

```
}

int main(){
    int n;
    cin>>n;
    myarray<telephone>address(100);
    for(int i=0;i<n;i++){
        int m;
        cin>>m;
        if(m==0){
            telephone temp;
            cin>>temp.name>>temp.tele>>temp.clas>>temp.domi;
            address.insert(temp);
        }
        else if(m==1){
            telephone temp;
            cin>>temp.name;
            address.erase(temp);
        }
        else if(m==2){
            telephone temp;
            int p;
            string t;
            cin>>temp.name>>p>>t;
            address.change(temp,p,t);
        }
        else if(m==3){
            telephone temp;
            cin>>temp.name;
            address.find(temp);
        }
        else{
            string t;
            cin>>t;
            address.output(t);
        }
    }
}
```