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：201805130155 | 姓名： 赵雨晗 | | 班级： 18计科3班 |
| 实验题目：数组描述线性表 | | | |
| 实验学时：2 | | 实验日期： 2019.9.25 | |
| 实验目的：  1、 掌握线性表结构、数组描述方法（顺序存储结构）、数组描述线性表的实现。  2、 掌握线性表应用。 | | | |
| 软件开发工具：  Windows : Vs Code + MingW | | | |
| 1. 实验内容   1、 创建线性表类：线性表的存储结构使用数组描述，提供操作: 插入、删除、查找等。  2、 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。由键盘输入或文件录入的通讯录信息建立通讯录表，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所有人信息。     1. 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）   使用线性表，写一个不定长的数组。   1. 测试结果（测试输入，测试输出）      1. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）   一开始一直WA，后来发现在使用了ios::sync\_with\_stdio的同时使用scanf造成了错误  并无问题   1. 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 <cstdio>  #include <iostream>  #include <string>  using namespace std;  struct node{  string name;  long long tele;  long long \_class;  long long \_domi;  };  template <class T>  class ArrayList{  protected:  T\* \_Ele;  int Arr\_len;  int list\_size;  public:  class iterator;  iterator begin(){  return iterator(\_Ele);  }  iterator end() {  return iterator(\_Ele + Arr\_len);  }  class iterator{  protected:  T\* position;  public:  typedef bidirectional\_iterator\_tag iterator\_category;  typedef T value\_type;  typedef ptrdiff\_t difference\_type;  typedef T\* pointer;  typedef T& reference;  iterator(T\* thePosition = NULL){position = thePosition;}  T& operator\*() const{return \*position;}  T\* operator->() const{return &\*position;}  iterator &operator++(){  ++position;return \*this;  }  iterator operator++(int){  iterator old = \*this;  ++position;  return old;  }  iterator& operator--(){  --position; return \*this;  }  iterator operator--(int){  iterator old = \*this;  --position;  return old;  }  };  ArrayList(int init\_L = 10);  ArrayList(const ArrayList<T>&);  void push\_back(const T &x);  void output();  int checkIndex(int x){  return x < Arr\_len;  }  int del(int x){  if (!checkIndex(x)) return 0;  for (int i = x + 1; i < Arr\_len; ++i)  \_Ele[i - 1] = \_Ele[i];  Arr\_len--;  return 1;  }  T operator [](const int &t){return \*(\_Ele + t);}  int size(){  return Arr\_len;  }  void change(int x, T a){  \_Ele[x] = a;  }  };  template <class T>  ArrayList<T> :: ArrayList(int init\_L){  list\_size = init\_L;  \_Ele = new T[init\_L];  Arr\_len = 0;  };  template <class T>  ArrayList<T> ::ArrayList(const ArrayList<T> &t) {  list\_size = t.list\_size;  \_Ele = new T[list\_size];  Arr\_len = t.Arr\_len;  copy(t.\_Ele, t.\_Ele + t.Arr\_len, \_Ele);  }  template <class T>  void ArrayList<T> ::push\_back(const T &x) {  if (list\_size == Arr\_len) {  ArrayList<T>A(\*this);  list\_size = list\_size \* 2;  delete[] \_Ele;  \_Ele = new T[list\_size];  copy(A.\_Ele + 1, A.\_Ele + Arr\_len, \_Ele);  }  \_Ele[Arr\_len++] = x;  }  template <class T>  void ArrayList<T> ::output() {  for (int i = 0; i < Arr\_len - 1; ++i)cout<<\_Ele[i]<<' ';  cout<<\_Ele[Arr\_len - 1];  // cout<<endl;  }  int del\_name(ArrayList<node> &A, const string &t) {  int n = A.size(), \_x = -1;  for (int i = 0; i < n; ++i)  if (A[i].name == t){  \_x = i;  break;  }  if (\_x == -1) return 0;  else return A.del(\_x);  }  void \_modify(ArrayList<node> &A, const string &t, int x, long long val){  int n = A.size();  for (int i = 0; i < n; ++i){  if (A[i].name == t){  node a = A[i];  if (x == 1) a.tele = val;  else if (x == 2) a.\_class = val;  else a.\_domi = val;  A.change(i, a);  return;  }  }  }  int \_find(ArrayList<node> &A, const string &t){  int n = A.size();  for (int i = 0; i < n; ++i)  if (A[i].name == t) return 1;  return 0;  }  long long \_class\_xor(ArrayList<node> &A, const long long &val){  int n = A.size();  long long ans = 0;  for (int i = 0; i < n; ++i){  if (A[i].\_class == val)  ans ^= A[i].\_domi;  }  return ans;  }  int main()  {  ArrayList<node> \_L(20010);  int n, a;  string name;  long long \_class, \_domi, \_tele;  ios::sync\_with\_stdio(false);  cin>>n;  for (int i = 1; i <= n; ++i){  cin>>a;  if (a == 0){  cin>>name>>\_tele>>\_class>>\_domi;  \_L.push\_back((node){name, \_tele, \_class, \_domi});  } else if (a == 1){  cin>>name;  del\_name(\_L, name);  } else if (a == 2){  int x;  long long val;  cin>>name>>x>>val;  \_modify(\_L, name, x, val);  } else if (a == 3) {  cin>>name;  printf("%d\n", \_find(\_L, name));  } else {  cin>>\_class;  printf("%lld\n", \_class\_xor(\_L, \_class));  }  };  system("pause");  return 0;  } | | | |