|  |  |  |
| --- | --- | --- |
| **软件学院实验报告** | | |
| 姓名： 胡逢彬 学号： 2225060803 专业：网络工程 年级： 2022级 | | |
| 课程名称 | | 数据结构 |
| 实验名称 | | 实验1 集合的交、并、差 |
| 实验的准备阶段 | 实验内容 | （1）实验目的  通过该实验，让学生复习巩固C语言中的分支结构、循环结构、数组、链表、函数的调用等有关内容，体会到用数组存储集合时，需要记录集合元素的个数，否则输出结果会出现数据越界现象。  （2）实验内容  通过键盘，分别输入集合A和B中的数据元素，要求数据元素类型为整数类型，输出两个集合的交、并、差。  （3）实验要求  a)从程序完善性上考虑，集合元素输入时，要有检查元素重复的功能，每个集合中不允许有重复的元素。集合可以用数组存储，也可以用链表存储。  b)实现交、并、差运算时，分别把代码写成函数的形式，即实现交运算的函数，实现并运算的函数，实现差运算的函数，在主函数中分别调用三个函数。  c)使用菜单形式对应各个操作，应允许用户反复查看结果，想结束程序时，输入负数结束，使其编成一个完整的小软件。菜单参考示例如下：  1---输入集合A和B  2---求集合A交B  3---求集合A并B  4---求集合A-B  退出，输入一个负数！  菜单：建议不要做成清屏和刷屏的效果，否则后面截图会比较麻烦。  （4）验收/测试用例  如：  输入： A={1,2,3,4,5} B={3,4,5,6,7}  要注意输入的过程中，每输入一个元素都要检查输入的这个元素是否和前面的元素重复，如果重复，要求用户重新输入当前元素。验收测试时要测试这种重复的情况。  输出：A交B={3， 4， 5} A并B={1，2，3，4，5，6，7}  A-B={1, 2} |
| 实验类型 | 验证性 |
| 实验的重点、难点 | 重点： 数组或线性表的使用  难点： 去重操作和函数调用 |
| 实验环境 | VC++6.0 或 DEV |
| 实验的实施阶段 | 实验步骤及完成任务情况 | 一、设计思想  1.查重或去重  通过时间复杂度为O(n)的遍历，寻找是否有与添加数字相同的数，若有则非法，若无则加入新的数字；   1. 交   时间复杂度为O(mn)，对每个集合A中的元素都进行O(m)判断是否在B中出现过   1. 并   时间复杂度为 O(2000),通过桶排，记录集合A 和 集合B的出现元素并在排序后输出   1. 差   时间复杂度为O(mn)，对每个集合A中的元素都进行O(m)判断是否在B中出现过  二、主要源代码  #include<iostream>  constexpr int N=1e6+10;  int a[N];  int b[N];  int vis[N];  int n=0;  int m=0;  void add()  {  std::cout<<"输入集合A的数字（不可重复，输入-1结束）\n";  while(1)  {  int x;  std::cin>>x;  if(x==-1)break;  int flag=0;  for(int i=0;i<n;i++)  {  if(a[i]==x)  {  flag=1;  break;  }  }  if(flag==1)  {  std::cout<<"A集合输入重复\n";  }  if(flag==0)  {  a[n]=x;  n++;  }  }  std::cout<<"输入集合B的数字（不可重复，输入-1结束）\n";  while(1)  {  int x;  std::cin>>x;  if(x==-1)break;  int flag=0;  for(int i=0;i<m;i++)  {  if(b[i]==x)  {  flag=1;  break;  }  }  if(flag==1)  {  std::cout<<"B集合输入重复\n";  }  if(flag==0)  {  b[m]=x;  m++;  }  }  std::cout<<"读入结束\n";  }  void sub()  {  std::cout<<"输出A交B：\n";  for(int i=0;i<n;i++)  {  for(int j=0;j<m;j++)  {  if(a[i]==b[j])  {  std::cout<<a[i]<<" ";  }  }  }  std::cout<<"\n";  }  void sum()  {  for(int i=-1000;i<=1000;i++)  {  vis[i+1000]=0;  }  for(int i=0;i<n;i++)  {  vis[a[i]+1000]=1;  }  for(int i=0;i<m;i++)  {  vis[b[i]+1000]=1;  }  std::cout<<"输出A并B: ";  for(int i=-1000;i<=1000;i++)  {  if(vis[i+1000]==1)std::cout<<i<<" ";  }  std::cout<<"\n";  }  void cut()  {  std::cout<<"输出A-B: ";  for(int i=0;i<n;i++)  {  int flag=0;  for(int j=0;j<m;j++)  {  if(a[i]==b[j])  {  flag=1;  break;  }  }  if(flag==0)  {  cout<<a[i]<<' ';  }  }  cout<<"\n";  }  void print\_setA()  {  std::cout<<"输出集合A: ";  for(int i=0;i<n;i++)  {  std::cout<<a[i]<<" ";  }  std::cout<<"\n";  }  void print\_setB()  {  std::cout<<"输出集合B: ";  for(int i=0;i<m;i++)  {  std::cout<<b[i]<<" ";  }  std::cout<<"\n";  }  void menu()  {  std::cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";  std::cout<<"\*\*1---输入集合A和B \*\*\n";  std::cout<<"\*\*2---求集合A交B \*\*\n";  std::cout<<"\*\*3---求集合A并B \*\*\n";  std::cout<<"\*\*4---求集合A-B \*\*\n";  std::cout<<"\*\*5---输出集合A \*\*\n";  std::cout<<"\*\*6---输出集合B \*\*\n";  std::cout<<"\*\* 输入-1退出 \*\*\n";  std::cout<<"\*\*请您选择（输入数字）：\n";  int op;  std::cin>>op;  if(op==1){  add();  }  else if(op==2){  sub();  }  else if(op==3)  {  sum();  }  else if(op==4)  {  cut();  }  else if(op==5)  {  print\_setA();  }  else if(op==6)  {  print\_setB();  }  else if(op==-1)  {  exit(0);  }  system("pause");  system("cls");  }  int main()  {  while(1)  {  menu();  }  return 0;  } |
| 实验结果的处理阶段 | 实验结果 | 16969936185301696993649269  16969936916801696993712633  169699374789616969937647551696993771124 |
| 实验结果总结 | 问题1：桶排时，由于考虑没考虑负数，数组下标出现负数，导致数据异常  解决方法：通过增加一个偏移量，使负数也可以进行正常操作  问题2：许多输出语句堆积，页面不简洁，于是增加system("cls")  导致原本想要的输出语句被清空  解决方法：通过增加语句  system("pause");  先按任意键再清空 |