山东大学 软件 学院

c++程序设计 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301249 | 姓名：王帅 | | 班级：软件4班 |
| 实验题目：实验二 | | | |
| 实验学时：4 | | 实验日期： 2019.4.10 | |
| 实验目的：  熟悉c++的数组。  熟悉c++的指针、引用。 | | | |
| 硬件环境：  PC机 | | | |
| 软件环境：  DEV C++ | | | |
| 实验步骤与内容：  实验三：  源程序文件myFunctions.cpp：  --begin—c—code--  int findMaxElement1D(int x[],int n) //返回一维数组中的最大值  {  int max=x[0];  for(int i=1;i<n;i++)  {  if(x[i]>max)  max=x[i];  }  return max;  }  int findMaxElement2D(int x[][5],int r) //返回二维数组中的最大值  {  int max=x[0][0];  for(int m=0;m<r;m++)  {  for(int n=0;n<5;n++)  {  if(x[m][n]>max)  max=x[m][n];  }  }  return max;  }  void sort1D(int x[],int n) //对一维数组用冒泡法排序  {  for (int j=0; j<n-1; j++)  {  bool s=true;  for (int i=0; i<n-1-j; i++)  {  if (x[i]>x[i+1])  {  s=false;  int t=x[i];  x[i]=x[i+1];  x[i+1]=t;  }  }  if(s) break;  }  }  --end—c—code--  头文件myFunctions.h，声明上述3个函数：  --begin—c—code--  #include "myFunctions.cpp"  int findMaxElement1D(int x[],int n);  int findMaxElement2D(int x[][5],int r);  void sort1D(int x[],int n);  --end—c—code--  定义主函数main()：  --begin—c—code--  #include "myFunctions.h"  #include <iostream>  using namespace std;  int main()  {  int x1[10];  int x2[3][5];    cout<<"请输入一维数组："<<endl;  for(int m=0;m<10;m++)  {  cin>>x1[m];  }    cout<<"请输入二维数组："<<endl;  for(int m=0;m<3;m++)  for(int n=0;n<5;n++)  {  cin>>x2[m][n];  }    cout<<endl<<"输入的一维数组："<<endl;  for(int m=0;m<10;m++)  {  cout<<x1[m]<<" ";  }    cout<<endl<<"输入的二维数组："<<endl;  for(int m=0;m<3;m++)  {  for(int n=0;n<5;n++)  {  cout<<x2[m][n]<<" ";  }  cout<<"\n";  }    cout<<"一维数组最大值："<<endl;  int m1 = findMaxElement1D(x1,10);  cout<<m1<<endl;    cout<<"二维数组最大值："<<endl;  int m2 = findMaxElement2D(x2,3);  cout<<m2<<endl;    cout<<"排序后的一维数组："<<endl;  sort1D(x1,10);  for(int m=0;m<10;m++)  {  cout<<x1[m]<<" ";  }    return 0;  }  --end—c—code--  实验结果如下：    实验四：  第一题代码：  --begin—c—code--  #include <iostream>  using namespace std;  void copy(char \*p,char \*q,int m)  {  p=p+m-1;  while(\*p!='\0' ) {\*q++=\*p++;}  \*q!='\0';  }  int main()  {  char a[30],b[30],\*p,\*q;  int m;  cout<<"请输入一个长度小于30的字符串：";  cin>>a;  cout<<"请输入m的值:";  cin>>m;  p=a;  q=b;  copy(p,q,m);  cout<<b;  return 0;  }  --end—c—code--  输出结果：    第二题代码：  --begin—c—code--  void transform(int \*p)  {  int t;  t=\*(p+1); \*(p+1)=\*(p+3); \*(p+3)=t;//矩阵转置操作  t=\*(p+2); \*(p+2)=\*(p+6); \*(p+6)=t;  t=\*(p+5); \*(p+5)=\*(p+7); \*(p+7)=t;  }  int main()  {  int a[9],i,r=0,\*p;    cout<<"请输入矩阵元素："<<endl;    for(i=0;i<9;i++) {cin>>a[i];}    cout<<"原矩阵："<<endl;    for(i=0;i<9;i++)  {  if(r%3==0&&r!=0) {cout<<endl;}  cout<<a[i]<<" ";  r++;  }  cout<<endl;    p=a;  transform(p);    cout<<"转制后矩阵："<<endl;    for(i=0;i<9;i++)  {  if(r%3==0&&r!=9) {cout<<endl;}  cout<<a[i]<<" ";  r++;  }    return 0;  }  --end—c—code--  输出结果：    第三题代码：  --begin—c—code--  #include <iostream>  using namespace std;  void getnumber(char \*p,int length)  {  int a[100],m,n,number,index=0;    for(m=0;m<length;m++)  {  if(\*(p+m)>='0'&&\*(p+m)<='9') //判断是否为数字  {  number=\*(p+m)-'0';  for(n=m+1;n<=length;n++)  {  if(\*(p+n)>='0'&&\*(p+n)<='9')//判断下一个是否为数字 {number=number\*10+\*(p+n)-'0'; m++;}  else break;  }  a[index]=number;  index++;  }  }  cout<<"存放数字的数组为: "<<endl;  for(int i=0;i<index;i++) {cout<<a[i]<<" ";}  }  int main()  {  char x[100];  int length;  cout<<"请输入一个字符串："<<endl;  cin>>x;  cout<<endl;  length=sizeof(x)-1;  getnumber(x,length);    return 0;  }  --end—c—code--  输出结果： | | | |
| 结论分析与体会：  C++的数组和指针操作多我来说的确有一些困难，和java相比有较大的区别，需要更加深入地理解C++编程语言的思想,只有将指针和引用充分理解，才能较为熟练地应用之。 | | | |