**暨南大学本科实验报告专用纸**

课程名称 高级语言程序设计实验 成绩评定

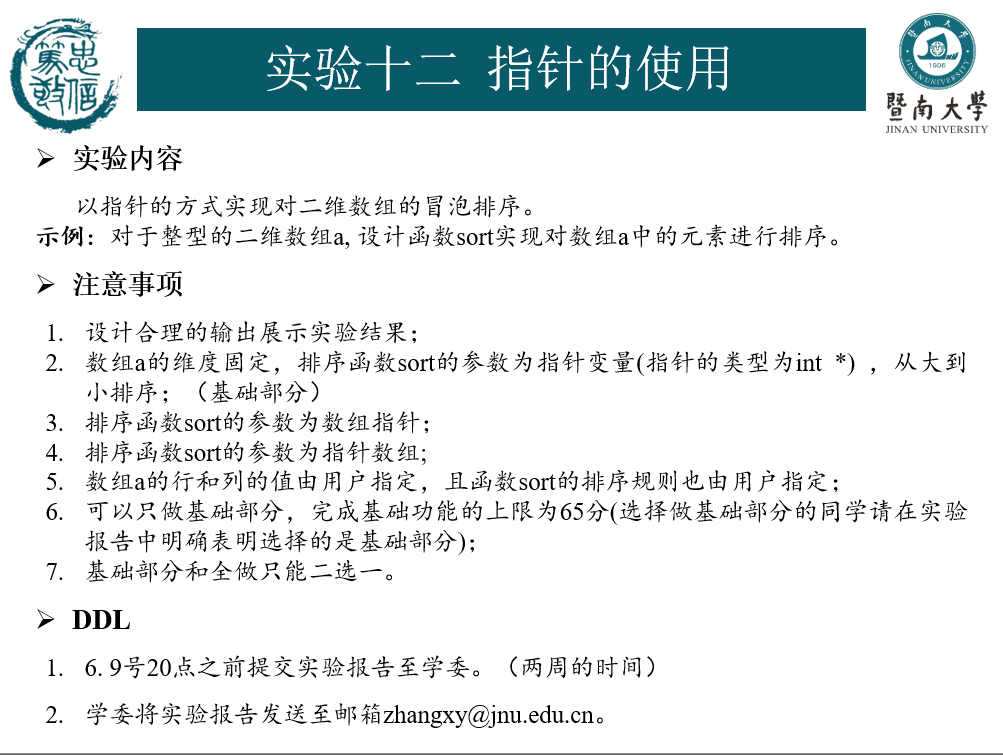
实验项目名称 指针的使用 指导教师 张鑫源

实验项目编号 110 实验项目类型编程实验 实验地点

学生姓名 林晓旭 学号 2019051121

智能工程与科学学院 系 信息安全 专业

实验时间2020 年—5月30日 上午～5月30日下午 温度 ℃湿度

1. **实验目的：熟悉c语言中指针，指针数组，数组指针的使用**
2. **实验内容和要求：  
   **
3. **主要仪器设备：计算机**

**（四） 源程序：**

**#include<iostream>**

**#include<ctime>**

**#include<cstdlib>**

**using namespace std;**

**int row,column,b;**

**void print(int\*\* p)//打印数组**

**{**

**for(int i=0;i<row;i++)**

**for(int j=0;j<column;j++)**

**cout<<p[i][j]<<" ";**

**cout<<endl;**

**}**

**void print(int (\*p)[100])**

**{**

**for(int i=0;i<row;i++)**

**for(int j=0;j<column;j++)**

**cout<<(p[i])[j]<<" ";**

**cout<<endl;**

**}**

**void free\_memory(int \*\*p)//释放动态分配的内存**

**{**

**for(int i=0;i<row;i++)**

**delete p[i];**

**delete p;**

**}**

**bool compare(int x,int y,int i)// 自定义偏序关系**

**{**

**if(i==1){**

**if(x>y)return true;**

**else return false;**

**}**

**else if(i==2)**

**{**

**if(x<y)return true;**

**else return false;**

**}**

**}**

**void my\_sort(int \*\*p)//以int类型的二级指针作为参数以及数组指针作为传递参数**

**{**

**for(int k=row\*column;k>0;k--)**

**{**

**bool flag=true;**

**for(int i=0;i<row;i++)**

**{**

**if(!flag)break;**

**for(int j=0;j<column;j++)**

**{**

**if(i\*column+j>=k){flag=false;break;**

**}**

**if(j==column-1&&i!=row-1){**

**if(compare(p[i][j],p[i+1][0],b)){**

**int t=p[i][j];**

**p[i][j]=p[i+1][0];**

**p[i+1][0]=t;**

**}**

**}**

**else if(j!=column-1){**

**if(compare(p[i][j],p[i][j+1],b)){**

**int t=p[i][j];**

**p[i][j]=p[i][j+1];**

**p[i][j+1]=t;**

**}**

**}**

**}**

**}**

**}**

**print(p);**

**}**

**void my\_sort(int (\*p)[100])//以数组指针作为参数**

**{**

**for(int k=row\*column;k>0;k--)**

**{**

**bool flag=true;**

**for(int i=0;i<row;i++)**

**{**

**if(!flag)break;**

**for(int j=0;j<column;j++)**

**{**

**if(i\*column+j>=k){flag=false;break;**

**}**

**if(j==column-1&&i!=row-1){**

**if(compare(p[i][j],p[i+1][0],b)){**

**int t=p[i][j];**

**p[i][j]=p[i+1][0];**

**p[i+1][0]=t;**

**}**

**}**

**else if(j!=column-1){**

**if(compare(p[i][j],p[i][j+1],b)){**

**int t=p[i][j];**

**p[i][j]=p[i][j+1];**

**p[i][j+1]=t;**

**}**

**}**

**}**

**}**

**}**

**print(p);**

**}**

**int main(void)**

**{**

**int t=3;**

**while(t--)**

**{**

**srand(time(NULL));//设定产生随机数的种子**

**cout<<"请输入要生成的二维数组的行数和列数："<<endl;**

**cin>>row>>column;**

**cout<<"按升序排序请输入1，按降序排序请输入2。"<<endl;**

**cin>>b;**

**if(t!=0){**

**int \*\*a=new int\*[row];**

**for(int i=0;i<row;i++)**

**a[i]=new int[column];**

**for(int i=0;i<row;i++)**

**for(int j=0;j<column;j++)**

**a[i][j]=rand()%100;**

**cout<<"随机生成的数组为："<<endl;**

**for(int i=0;i<row;i++)**

**{**

**for(int j=0;j<column;j++)**

**cout<<a[i][j]<<" ";**

**cout<<endl;**

**}**

**cout<<"排序后的数组为："<<endl;**

**if(t==2)my\_sort(a);**

**else if(t==1)my\_sort(a);**

**free\_memory(a);**

**}**

**else{**

**int (\*c)[100]=new int[row][100];**

**for(int i=0;i<row;i++)**

**for(int j=0;j<column;j++)**

**c[i][j]=rand()%100;**

**cout<<"随机生成的数组为："<<endl;**

**for(int i=0;i<row;i++)**

**{**

**for(int j=0;j<column;j++)**

**cout<<c[i][j]<<" ";**

**cout<<endl;**

**}**

**cout<<"排序后的数组为："<<endl;**

**my\_sort(c);**

**delete c;**

**}**

**}**

**}**

**（五）、步骤与调试**

**1、读取用户输入的行数row和列数column，以及排序方式(升序或降序)。**

**2、根据用户制定的行数和列数动态的分配内存空间来创建一个row×column的整型二维数组，在使用srand()函数设定随机种子后使用rand()函数来给数组赋初值。**

**3、compare()函数来实现根据输入的不同来不同地排序。**

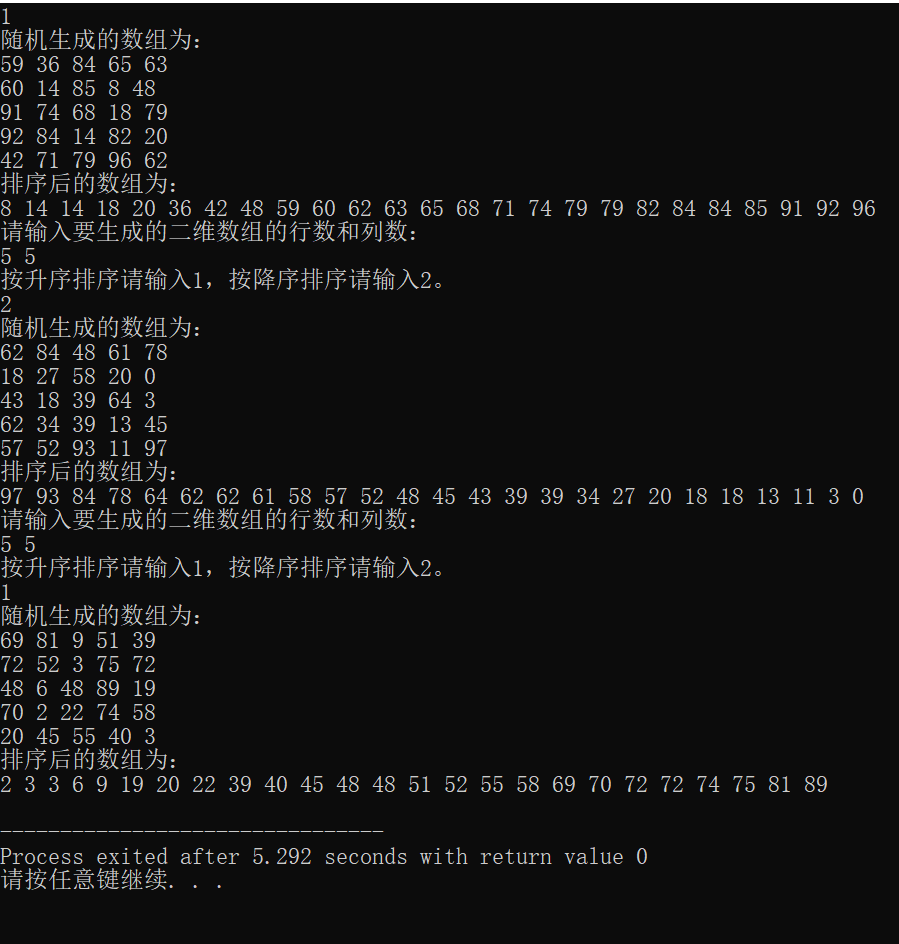
**4、使用my\_sort()函数(函数重载)来给二维数组里面的元素排序。**

**5、用print()函数(函数重载)将数组排成一行输出。**

**6、用delete删除分配的内存。**

## （六）、实验结果与分析

**1、程序输出结果：**

****

**2、分析：**

**(1)、srand()函数是随机数发生器的初始化函数，通过参数seed来生成随机种子，相同的随机种子生成的随机数数列是相同的，所以可以将time()函数的返回值作为随机种子参数，让每次程序的运行生成的随机数不同；rand()函数的作用是生成随机数；通过计算机生成的随机数是伪随机数，但因为符合正态分布且循环周期长，在一般条件下可将其看成是随机数。**

**(2)、程序中通过new关键字来动态地分配内存给数组，不会造成内存的浪费，生成动态二维数组的方法有两种。比如生成一个row×column的数组：**

**第一种方法：**

**int \*\*a=new int\*[row];**

**for(int i=0;i<row;i++)**

**a[i]=new int[column];**

**先定义一个指向int类型的二级指针，让其指向一个int类型的一级指针数组，长度为row，该指针数组中的每个元素又指向一个长度为column的整型数组。**

**第二种方法：**

**int (\*c)[100]=new int[row][100];**

**定义一个数组指针，因为数组指针一定要规定其指向的数组的长度(这就像定义变量时要明确指明变量的类型)，所以没办法按照用于的输入来规定生成数组的列数。但是可以规定一个较大的列数来满足用户的需求(虽然一定程度上造成了内存浪费，但是避免因数组越界造成程序崩溃).**

**(3)、通过循环三次调用my\_sort函数，排序算法都是冒泡排序，但依次实现了以指针、指针数组、数组指针作为传递参数。本程序利用了c++支持函数重载的特性，根据不同的参数表调用内容不同的同名函数，免去了多次命名造成程序结构的混乱。在这个过程中发现了程序将以下两个函数看成了同一个函数**

**void my\_sort(int \*\*p)；**

**void my\_sort(int (\*p)[100])**；

**这说明了两个函数参数的本质是相同的，在第二个函数中p的实质是一个指向int类型的二级指针。这是因为在传递参数的过程中，实参是数组指针的地址，在调用函数时，程序会生成指针p指向数组指针的地址，所以p实质上是指向指针的指针也就是二级指针。**

**这和**

**void function(int\* p)**

**void function(int p[])**

**是一个道理；至于为什么使用指向int类型的二级指针而不是指向int类型的一级指针，是因为对于上述第一种方法动态分配的二维数组，其内存不是线性连续的(即列最后一个元素的地址加一就是下一行第一个元素的地址)，所以不可以单单通过循环递增一个int类型的指针来实现数组的遍历；但是对于第二种方法动态声明的二维数组以及普通声明的二维数组却可以实现只通过递增一个int类型指针实现遍历。**

**（4）、通过compare()函数实现自定义元素的偏序关系，相当于重载了比较运算符。**

**bool compare(int x,int y,int i)// 自定义偏序关系**

**{**

**if(i==1){**

**if(x>y)return true;**

**else return false;**

**}**

**else if(i==2)**

**{**

**if(x<y)return true;**

**else return false;**

**}**

**}**

**（5）、通过free\_memory()函数释放动态分配的内存，以免内存泄漏。**

**暨南大学本科实验报告专用纸(附页)**