山东大学 软件 学院

数据结构 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目：递归练习 | | | |
| 实验学时：4 | | 实验日期： | |
| 实验目的：  1、 熟悉开发工具的使用。  2、 掌握递归的实现思想。 | | | |
| 硬件环境：  lntel(R) Core(TM) i5-10210U CPU @ 1.60G Hz 2 .1 1 G Hz | | | |
| 软件环境：  CLion 2020.3.3 x64 | | | |
| 实验步骤与内容：  实验内容：  1、 输入2-20个大于0的正整数（1、2、3或者100、200、300），输入0作为结束，0不参与排列。  2、 输出这几个整数的全排列，每个数之间用半角“，”隔开，中间不要有空格，每个排列单独一行。  3、 程序一定要有Input、Output、End0(End后可以加一个0-9的数字代表版本，当出现结果错误，QQ咨询老师，改变一下版本，目的就是确保上传到平台的程序和自己测试的是同一个版本的程序)提示信息，但是不要有格式没有出现的其他提示，以下各实验相同。  实验步骤：  1.安装实验环境：clion+MingGW，测试实验环境正常  2.复习相关知识：递归、全排列  3.编写代码 | | | |
| 结论分析与体会：  1.递归：正确的递归函数必须包含基础部分。每一次递归调用，其参数值都比上一次的参数值要小，从而重复调用递归函数参数值达到基础部分的值。  2.输出n个元素的全排列，用递归函数编写比较容易，当n=1时是递归的基础部分，这时集合只有一个元素；当n>1时，成为递归部分，组成了一个完整的递归定义。  3.要注意的是使用递归函数时交换的两个数值分别是当前选定的特定元素和这个元素后面的每一个元素，这样才能实现准确的全排列。 | | | |

**本次实验主要代码：**

**//打印出结果**

void givRes(int arr[],int len){

for (int i = 0; i < len; ++i) {

if (i==(len-1))cout << arr[i]<<endl;

else cout<<arr[i]<<",";

}}

//递归进行全排列

void giveRage(int arrays[],int begLen,int endLen){

if (begLen == endLen)givRes(arrays,endLen);

else{

for (int i = begLen; i < endLen; ++i) {

swap(arrays[begLen], arrays[i]);

giveRage(arrays,begLen+1,endLen);

swap(arrays[begLen], arrays[i]);

}

}

}

//程序入口

int main() {

int nums[20],count=0;

cout << "Input" << endl;

for (int i = 0; i < 20; ++i) {

int temp;

cin>>temp;

if (temp==0)break;

else {

nums[i] = temp;

count++;

}

}

cout << "Output" << endl;

giveRage(nums,0,count);

cout << "End0"<<endl;

return 0;

}