山东大学 软件 学院

数据结构 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：  1、编写一个C++递归函数，输出n个元素的全排列。  2、编写一个C++递归函数，输出n个元素的所有子集。 | | | |
| 实验学时：60分钟 | | 实验日期： 2018.9.14 | |
| 实验目的：熟悉c++的递归用法。 | | | |
| 硬件环境： 笔记本电脑 | | | |
| 软件环境：win10+Visual Studio 2017 | | | |
| 实验内容：  1、基本代码如下：  #include <iostream>  #include<iterator>  using namespace std;  template <typename T>  void permutations(T list[], int k, int m)  {  if (k == m)  {  copy(list, list + m + 1, ostream\_iterator <T>(cout, ""));  cout << endl;  }  else  for (int i = k; i <= m; i++) {  swap(list[k], list[i]);  permutations(list, k + 1, m);  swap(list[k], list[i]);  }  }  int main()  {  char a[3] = { 'a','b','c' };  permutations(a,0,2);  }  2、基本代码如下：  #include<iostream>  #include<vector>  using namespace std;  template<typename T>  void Subset(const T vec[], int n, vector<T> set) {  if (n <= 0) {  cout << "{ ";  for (int i = 0; i < set.size(); i++)  cout << set[i];  cout << " }" << endl;  return;  }  Subset(vec + 1, n - 1, set);  set.push\_back(vec[0]);  Subset(vec + 1, n - 1, set);  }  int main() {  char vec[] = { 'a', 'b', 'c' };  vector<char> set;  Subset(vec, sizeof(vec), set);  return 0;  } | | | |
| 运行结果：  1、  本题课本已有代码，只需将其具体化即可。    2、  本题跟上一题有点类似，都是先给出一个函数模板，然后调用模板。  图片包含 屏幕截图  已生成极高可信度的说明 | | | |