《数据结构》实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 李爽 | 班级 | 22级软件工程专升本1班 | 学号 | 2206831522 |
| 实验名称 | **实验四 字符串和数组** | | | | |
| 实验时间 | 2023.4.27 | | 成绩 | |  |
| 1. **实验目的：**   1.掌握串的顺序存储结构；  2.验证顺序串及其基本操作的实现；  3.掌握串的操作特点,掌握数组的寻址方式。  **二、实验工具：**  软件平台：Windows7或以上版本，Visual Studio 2019  **三、实验原理：**  1、算法描述：  （1）求字符串长度函数：该函数输入一个字符串，通过遍历字符串的每个字符，统计字符的个数来计算出字符串的长度。  （2）字符串拼接函数：该函数输入两个字符串，将第二个字符串的每个字符按顺序添加到第一个字符串的末尾，最终形成一个新的字符串。  （3）比较大小函数：该函数输入两个字符串，逐一比较两个字符串的每个字符，如果字符相同，则继续比较下一个字符；如果字符不同，则直接返回结果。如果遍历完一个字符串后，两个字符串的前面所有字符都相同，则比较字符串的长度，长度较大的字符串更大。  （4）奇数阶幻方函数：这个程序是用来生成奇数阶幻方的。幻方是一个由若干个排列整齐的数组成的正方形，在其中任意一横行、一纵行以及对角线上的数之和都相等。这个程序使用了一个二维数组来存放幻方的值，并通过循环填数的方式来生成幻方。  **四、实验步骤和内容：**  1.定义任意两个字符串，完成以下操作：  （1）编写求字符串长度函数，显示该字符串长度；  （2）编写字符串拼接函数，将该字符串与键盘输入的另一字符串拼接，并显示拼接结果；  （3）编写比较大小函数，比较两字符串大小，并显示比较结果。  2.实现奇数阶幻方。在一个由若干个排列整齐的数组成的正方形中，图中任意一横行、一纵行的几个数之和都相等，具有这种性质的图表，称为“幻方”。按目前填写幻方的方法，是把幻方分成了三类，即奇数阶幻方、双偶阶幻方、单偶阶幻方。奇数阶幻方的方法比较简单，这里只考查奇数阶幻方的程序。输入30以内的任一奇数n，自动输出n\*n阶幻方。如：  输入示例：  3 5 7  输出示例：  6 1 8 15 8 1 24 17 28 19 10 1 48 39 30  7 5 3 16 14 7 5 23 29 27 18 9 7 47 38  2 9 4 22 20 13 6 4 37 35 26 17 8 6 46  3 21 19 12 10 45 36 34 25 16 14 5  9 2 25 18 11 4 44 42 33 24 15 13  12 3 43 41 32 23 21  20 11 2 49 40 31 22   1. 代码如下   #include <iostream>  #include <cstring>  using namespace std;  // 求字符串长度  int getLength(const char\* str) {  int length = 0;  //遍历获取长度  while (\*str != '\0') {  length++;  str++;  }  return length;  }  // 字符串拼接  void concatenate(char\* str1, char\* str2) {  strcat(str1, str2);  cout << "拼接结果：" << str1 << endl;  }  // 比较大小  void compare(char\* str1, char\* str2) {  int result = strcmp(str1, str2);  if (result == 0) {  cout << "两个字符串相等" << endl;  }  else if (result > 0) {  cout << "第一个字符串大于第二个字符串" << endl;  }  else {  cout << "第一个字符串小于第二个字符串" << endl;  }  }  int main() {  // 系统定义的字符串  char str1[] = "Hello, world!";  cout << "系统定义的字符串是：" << str1 << endl;  // 用户输入的字符串  cout << "请输入另一个字符串：";  char str2[100];  cin >> str2;  cout << "您输入的字符串是：" << str2 << endl;  // 求字符串长度  int length1 = getLength(str1);  int length2 = getLength(str2);  cout << "系统定义的字符串长度为：" << length1 << endl;  cout << "您输入的字符串长度为：" << length2 << endl;  // 字符串拼接  char str3[200];  strcpy(str3, str1);  concatenate(str3, str2);  // 比较大小  compare(str1, str2);  return 0;  }    2.代码如下  #include <iostream>  #include <cstring>  using namespace std;  const int N = 31; // 定义常量N为31，用于存放幻方的大小  int a[N][N]; // 定义二维数组a存放幻方的值  int main()  {  // 定义n存放幻方的大小  int n;  // 输入幻方的大小  cout << "请输入幻方大小n（1<= n <= 30）：";  cin >> n;  // 将二维数组a全部置0  memset(a, 0, sizeof a);  // 定义初始位置为第一行的中间位置  int x = n / 2, y = n - 1;  // 将1填入初始位置  a[x][y] = 1;  // 从2开始循环填充幻方  for (int i = 2; i <= n \* n; i++) {  // 计算下一个位置  int nx = (x - 1 + n) % n, ny = (y + 1) % n;  // 如果下一个位置已经填有数  if (a[nx][ny]) {  // 则填在当前位置的下面  x = (x + 1) % n;  } else {  // 否则填在计算出来的下一个位置  x = nx, y = ny;  }  // 将当前数填入幻方中  a[x][y] = i;  }  // 循环输出幻方  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << a[i][j] << " ";  }  cout << endl;  }  return 0;  }    **五、实验总结：**  在这次实验中，我使用C++语言完成了串的顺序存储结构的实现，并验证了顺序串及其基本操作的实现。通过实验，我不仅掌握了串的操作特点，还深入了解了数组的寻址方式。  在实验过程中，我首先学习了串的概念和特点，以及串的存储结构，进而了解了顺序串的实现原理。然后，我按照课程要求，使用C++语言编写了串的顺序存储结构的代码，并进行了测试验证。在测试过程中，我实现了顺序串的基本操作，例如串的赋值、求长度、串的拼接、串的截取、串的比较等，通过测试，我验证了顺序串及其基本操作的实现。  在实验过程中，我还了解了数组的寻址方式，包括数组的下标寻址和指针寻址。在顺序串的实现中，我使用了数组的下标寻址方式来访问串的各个元素。这个过程让我更加深入地了解了数组的寻址方式，掌握了数组的使用技巧。  通过这次实验，我不仅学会了如何实现顺序串及其基本操作，还深入了解了数组的寻址方式，这对我的编程技能和算法设计能力都有很大的提升。在以后的编程实践中，我一定会更加熟练地运用这些知识和技能，为我未来的学习和工作打下坚实的基础。  **六、教师评语：** | | | | | |