



计算机与软件工程学院

上机实验报告

**（ 2019/2020 学年 第 2 学期 ）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | C/C++程序设计 | | | |
| 课程代码 | 190901019 | | | |
| 任课教师 | 王晓明 | | | |
| 学生姓名 | 刘唐 | 年级 | 19级 |
| 学号 | 3120190971181 | 专业 | 计算机类 |
| 实验成绩 |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 实验2：函数与数组 | | | 实验地点 | Home |
| 实验类型 | 设计 | 实验学时 | 2 | 实验日期 | 2020/6/2 |
| * 撰写注意：版面格式已设置好（不得更改），填入内容即可。 | | | | | |
| 1. 实验目的   掌握自编函数的定义、库函数的调用、数组的定义与下标访问。 | | | | | |
| 1. 实验内容 2. 实验任务1   //给一个字符串 S, 你可以通过在前面添加字符将其转换为回文串.找到并返回用这种方式转换的最短回文串.  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   任意字符串**！【长度<50的任意内容 除了汉字😅】**  但是：**可以有回文内容甚至本身就是一个回文**   1. **数据存储**（输入数据在内存中的存储）   一个字符数组存储输入数据  一个temp数组  一个仿制指针（就是数组的下标，但是是独立存储在一个整数位置）   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   算法说明：输入字符串a，判断其是否回文，若不是：  把字符串最后一个字符复制后，放到原字符串-1的位置（原来的a字符串第一个字符前面而不是组合后的字符串前面）。直到判断是回文  输入数组a  用仿制指针对准最后一个位置i（任意长度嘛）  从a的i-1位开始往下  每次都判断已经接好的数组是否回文  否：把a的i-1位写到temp数组里面，再把原来的数组接上去  是：是回文了那就搞完啦  输出接好的回文数组     1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）   **//蓝色为输入，黄色为输出，那个大字的箭头没对准**     1. 实验任务2   折半查找  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   一个数值   1. **数据存储**（输入数据在内存中的存储）   就一个整型位置   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   通过数组下标找到数组中间位置，对比此数值与输入数值大小  更大（或更小）  数组下标中的最大值（或最小值）不变，最小值（或最大值）位置变为刚才对比位置+1（或-1）的位置  重复对比直到最大位置数值等于最小位置数值  如果此数值与要查找的位置数值相等  输出这个位置  不相等  输出：error信息   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）   附：原数组a**：**a[10] = **{ 0,1,2,3,4,5,6,7,8,9 }**     1. 实验任务3   **有一个已排好序的数组，今输入一个数，要求按原来排序的规律将它*插入数组中*。**  **//输出原来的数组和插入好的数组。**  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   一个随机选定的整数   1. **数据存储**（输入数据在内存中的存储）   一个比原来数量大一个位的数组，一个输入位置   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   设：输入数为num  判断第i个数是否比num大[i从0到9]  若是（n[i]<num）：从最后一个数往后调，知道调到i位置并且写入num  若是（最后一个数都比num小）直接写入此位置  循环输出修改后数组     1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大） | | | | | |
| 1. 实验环境 2. 操作系统：Windows10 3. 开发工具：visual studio 2019 4. 实验设备：PC | | | | | |
| 1. 提交文档   提交本实验报告（电子版），文件名命名：实验X\_报告\_学号\_姓名.doc  教师将**批阅后（有分数）**的全体学生实验报告刻入一张光盘存档，保证光盘可读。 | | | | | |
| 1. 附：源代码   实验一：  //给一个字符串 S, 你可以通过在前面添加字符将其转换为回文串.找到并返回用这种方式转换的最短回文串.😄  #include<stdio.h>  #include<string.h>  #include<stdlib.h>  #include<iostream>  using namespace std;  int 回文判断函数(char a[100])  {  int i = 0;  for (i = 0; i < 100; i++)  {  if (a[i] == '\0')  {  break;  }  }  for (int j = 0; j <= i / 2; j++)  {  if (a[j] == a[i - 1])  {  i--;  }  else  {  return -1;  }  }  return 0;  }  int main()  {  char a[50] = { 0 }, b[100] = { 0 }, c[50] = { 0 }, d[100] = { 0 }, p = 0;  printf("给一个字符串 S, 通过在前面添加字符将其转换为回文串.找到并返回用这种方式转换的最短回文串.\n");  scanf\_s("%s", a, 50); //为了保证任意输入长度,因为c++不能实现回车下的任意长度输入。必须要~Z。所以用格式化输入  int i = 0;  for (i = 0; i <= 50; i++) //i=最后一个字符的位置  {  if (a[i] == '\0')  {  i -= 1;  break;  }  }  for (int n = 0;i>=0 ;i-- ,n++)//在最后一个位置前进行操作  {  strcpy\_s(c, 50, a);  strcpy\_s(d, 100, b); //c可能有点多余了，但是懒得再改了  strcat\_s(d, 100, c);  if (回文判断函数(d) != 0)  {  b[n] = a[i];  }  else if(回文判断函数(d) == 0)  {  strcat\_s(b, 100, a);  cout << b;  break;  }  }  cout << endl;  system("pause");  return 0;  }  实验二：  #include<iostream>  #include<stdio.h>  using namespace std;  int main()  {  int a[10] = { 0,1,2,3,4,5,6,7,8,9 }, x, min = 1, mid = 4, max = 9;    cin >> x;  for (;;)  {  if (a[mid] > x)  {  max = mid - 1; mid = (min + max) / 2;  }  else if (a[mid] < x)  {  min = mid + 1; mid = (min + max) / 2;  }  if (x == a[mid]) { printf("a[%d]=%d\n", mid, x); break; }  else if ((max - min == 1) & (mid != x)){cout << "未查找到该数据\n"; break;}  }  return 0;  }  实验三  #include <iostream>  using namespace std;  int main()  {  int n[10] = { 2, 27, 36, 66, 72, 77, 80, 88, 96 };  int num, i, j;  for (i = 0, cout << "旧数组： "; i < 9; )  cout << n[i++] << ' ';  cout << endl << "输入数值： ";  cin >> num;  for (i = 0; i < 9; i++)  {  if (num < n[i])  {  for (j = 9; j >= i; n[j] = n[j - 1], j--);  n[i] = num;  break;  }  else if (num > n[8])  {  n[9] = num;  break;  }  }  for (i = 0, cout << "新数组： "; i < 10; )  cout << n[i++] << ' ';  cout << endl;  system("pause");  return 0;  } | | | | | |
|  | | | | | |