|  |  |  |  |
| --- | --- | --- | --- |
| **学生学号** | 1023004416 | **实验课成绩** | 93 |

**武汉理工大学**

**学 生 实 验 报 告 书**

|  |  |
| --- | --- |
| **实验课程名称** | 高级语言程序设计 |
| **开课学院** | 计算机与人工智能学院 |
| **指导教师姓名** | 张蕊 |
| **学生姓名** | 高华辰 |
| **学生专业班级** | 计算机类2303 |

2023 — 2024 学年 第 一 学期

实验课程名称： 高级语言程序设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | 2023级实验4 运用数组、函数和指针进行程序设计 | | | 实验成绩 | 90 |
| 实验者 | 高华辰 | 专业班级 | 计算机类2303 | 组别 |  |
| 同组者 |  | | | 实验日期 | 2023年12月 18日 |
| 第一部分：实验分析与设计（可加页）   1. 实验内容描述（问题域描述）   学会运用函数、数组和指针进行程序设计。   1. 实验基本原理与设计（包括实验方案设计，实验手段的确定，试验步骤等，用硬件逻辑或者算法描述）   **6-3 两个4位正整数的后两位互换**  将输入的任意两个4位正整数的后两位互换，例如输入1234和5678，交换之后输出1278和5634。  **6-6 删除字符**  本题要求实现一个删除字符串中的指定字符的简单函数。  **6-8 使用函数实现字符串部分复制**  本题要求编写函数，将输入字符串t中从第m个字符开始的全部字符复制到字符串s中。  **7-1 字符串排序**  本题要求编写程序，读入5个字符串，按由小到大的顺序输出。  三、主要仪器设备及耗材  设备：机房电脑 | | | | | |

|  |
| --- |
| 第二部分：实验调试与结果分析（可加页）   1. 调试过程（包括调试方法描述、实验数据记录，实验现象记录，实验过程发现的问题等）   **6-3 两个4位正整数的后两位互换**  分析：本题应先将输入的两个数的后两位提取并存起来，再去抹掉输入的两个数的后两位并拼接上互相的后两位。  代码如下：  void fun(int \*p,int \*q)  {  int p1=\*p,q1=\*q;  p1%=100;q1%=100;  \*p=\*p-p1+q1;\*q=\*q-q1+p1;  }  **6-6 删除字符**  分析：本题可以在遍历过程中将非待删除字符按顺序覆写在原字符串中，并在结尾附上’\0’ 。  代码如下：  void delchar( char \*str, char c )  {  int i,position=0;  for(i=0;str[i]!='\0';i++)  {  if(str[i]!=c)  {str[position]=str[i];position++;}  }  str[position]='\0';  }  **6-8 使用函数实现字符串部分复制**  分析：本题可以在遍历过程中到达指定位置后开始将字符覆写到原字符串，并在结尾附上’\0’。  代码如下：  void strmcpy( char \*t, int m, char \*s )  {  int len;  for(len=0;t[len]!='\0';len++);  if(m-1>len) return;  else{  int i,j;  for(i=m-1,j=0;t[i]!='\0';i++,j++) s[j]=t[i];  s[j]='\0';  }  }  **7-1 字符串排序**  分析：本题的根本的逻辑仍然是排序，将strcmp函数作为冒泡排序中元素互换的判定条件即可。  代码如下：  #include<stdio.h>  #include<string.h>  char bubble(char s[5][80])  {  for(int i=0;i<4;i++)  {  for(int j=0;j<4-i;j++)  {  if(strcmp(s[j],s[j+1])>0)  {  char temp[80];  strcpy(temp,s[j]);  strcpy(s[j],s[j+1]);  strcpy(s[j+1],temp);  }  }  }  }   1. 实验小结、建议及体会   函数、数组、指针的结合可以完成许多复杂的任务。利用指针可以使函数具有多个返回值，完成更加复杂的数据处理。在遇到处理数组的问题时，利用指针可以将其代入函数中进行处理，进一步使代码模块化，逻辑更清晰。在之后面对更加复杂的数据结构时，函数、数组、指针的组合使用是极为重要的。编写大型程序时，要将功能分解为各个模块，为了保证主函数的可读性，函数、数组、指针需要结合得更加紧密，函数的命名也需要更规范。可以肯定的是，我们已经离设计完整的程序越来越近了。 |