

**实 验 报 告**

**（2023 / 2024 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 高级语言程序设计A（JS1031X0S） | | | | | |
| 实验名称 | 实验三 数组、指针与字符串实验 | | | | | |
| 实验时间 | 2023 | 年 | 12 | 月 | 14 | 日 |
| 指导单位 | 计算机学院、软件学院、网络空间安全学院 | | | | | |
| 指导教师 | 付竟芝 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 刘圣 | 班级学号 | B23051217 |
| 学院(系) | 自动化学院、人工智能学院 | 专 业 | 智能电网信息工程 |

| **实 验 报 告** | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **实验名称** | | **实验三 数组、指针与字符串实验** | | | | | | **指导教师** | | 付竟芝 | | |
| **实验类型** | | **综合** | | **实验学时** | | **2** | | **实验时间** | | 2023-12-14 | | |
| 1. **实验目的和要求**   （1）重点掌握一维数组的定义与使用，能通过函数的定义与调用实现一维数组中的常用和经典算法。会正确定义一级指针，理解其存储地址的实质，并且会利用一级指针来访问一维数组的元素。  （2）综合运用函数、数组和指针的知识，理解函数中的一维数组形式参数实质为一级指针变量形式参数，在调用时提供的实参数组名实质上传递的是数组首地址的参数传递（传地址）。  （3）理解值形式参数与指针形式参数的区别，通过跟踪调试理解传值与传地址的不同，及指针形参如何改变对应的实参变量。理解指针与指针所指向的内容之间的关系，理解指针间接引用的实质。  （4）正确使用字符数组及字符指针处理字符串。 | | | | | | | | | | | | |
| 二、**实验环境(实验设备)**  硬件: 微型计算机  软件: Windows 操作系统、Microsoft Visual Studio 2010 | | | | | | | | | | | | |
| **三、实验原理及内容**  **实验题目（1）【见实验教材实验五的题目１】：**编写程序exp5\_1.c，在主函数中定义一维数组int array[10]，自定义以下函数：输入数组元素，输出数组元素、求数组元素平均值、输出数组元素最大值、输出数组元素最小值、查找某数值元素是否存在（若存在，请输出下标）、给数组元素排序，要求在主函数中对各子函数进行调用和测试。  **实验解答：**   1. 写出完整的源程序代码并做适当注释：   /\*编写程序exp5\_1.c，在主函数中定义一维数组int array[10]，  自定义以下函数：输入数组元素，输出数组元素、求数组元素平均值、  输出数组元素最大值、输出数组元素最小值、查找某数值元素是否存在（若存在，请输出下标）、  给数组元素排序，要求在主函数中对各子函数进行调用和测试。\*/  #include <stdio.h>  #include <stdlib.h>  void input(int array[], int n); // 输入  void output(int array[], int n); // 输出  double average(int array[], int n); // 平均值  int max(int array[], int n); // 最大值  int min(int array[], int n); // 最小值  void find(int array[], int n); // 查找  void sort(int array[], int n); // 排序  int main()  {  int array[10];  input(array, 10);  output(array, 10);  double ave = average(array, 10);  int maxnum = max(array, 10);  int minnum = min(array, 10);  printf("平均值：%lf\n", ave);  printf("最大值：%d\n", maxnum);  printf("最小值：%d\n", minnum);  find(array, 10);  sort(array, 10);  return 0;  }  void input(int array[], int n)  {  int i;  for (i = 0; i < n; i++)  {  printf("请输入第%d个数：", i + 1);  scanf("%d", &array[i]);  }  }  void output(int array[], int n)  {  int i;  for (i = 0; i < n; i++)  {  printf("%d\t", array[i]);  }  printf("\n");  }  double average(int array[], int n)  {  int i, sum = 0;  double ave;  for (i = 0; i < n; i++)  {  sum += array[i];  }  ave = sum / (double)n;  return ave;  }  int max(int array[], int n)  {  int i, max = array[0];  for (i = 1; i < n; i++)  {  if (array[i] > max)  max = array[i];  }  return max;  }  int min(int array[], int n)  {  int i, min = array[0];  for (i = 1; i < n; i++)  {  if (array[i] < min)  min = array[i];  }  return min;  }  void find(int array[], int n)  {  int i, num;  printf("请输入要查找的数：");  scanf("%d", &num);  for (i = 0; i < n; i++)  {  if (array[i] == num)  {  printf("第%d个数是%d\n", i + 1, num);  break;  }  }  if (i == n)  printf("没有找到！\n");  }  void sort(int array[], int n)  {  int i, j, temp;  for (i = 0; i < n - 1; i++)  {  for (j = i + 1; j < n; j++)  {  if (array[i] > array[j])  {  temp = array[i];  array[i] = array[j];  array[j] = temp;  }  }  }  printf("排序后：");  output(array, n);  }  ② 运行一次程序，写出你输入的数据以及程序每步的输出结果(从键盘输入的内容用斜体字)：  请输入第1个数：10  请输入第2个数：9  请输入第3个数：8  请输入第4个数：7  请输入第5个数：6  请输入第6个数：5  请输入第7个数：4  请输入第8个数：3  请输入第9个数：2  请输入第10个数：1  10 9 8 7 6 5 4 3 2 1  平均值：5.500000  最大值：10  最小值：1  请输入要查找的数：5  第6个数是5  排序后：1 2 3 4 5 6 7 8 9 10  **实验题目（2）【见实验教材实验六的题目2】：**编程exp6\_2.c，现有整型变量x，y（初始化为1和2），调用交换函数以期实现两个值的交换。下表中4种不同的定义及调用分别进行测试并填写表格。  **表1 拟实现交换的四种方法**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 原型声明 | void swap1( int , int ); | void swap2( int \*, int ); | void swap3( int \*, int \*); | void swap4( int \*, int \*); | | 调用  语句 | swap1( x , y ); | swap2( &x , y ); | swap3( &x , &y ); | swap4( &x , &y ); | | 函数  定义 | void swap1(int a, int b )  { int temp= a;  a = b;  b = temp;  } | void swap2(int \*a, int b )  { int temp= \*a;  \*a = b;  b = temp;  } | void swap3(int \*a, int \*b )  { int temp = \*a;  \*a = \*b;  \*b = temp;  } | void swap4(int \*a, int \*b )  { int \*temp = a;  a=b;  b=temp;  } |   **实验解答：**  ① 主函数代码如下：  int main()  {  int x=1,y=2;  swap1(x,y); /\*此处依次换成另外3种调用进行观察，其余代码不变\*/  printf("x=%d,y=%d\n",x,y);  return 0;  }  通过修改以上主函数中调用函数的语句，按表1更新对应的函数调用，分别运行程序，填写下表：   |  |  |  |  | | --- | --- | --- | --- | | **函数原型** | **输出结果** | **是否交换** | **原因分析** | | void swap1( int a, int b ); | x=1,y=2 | 否 | 形式参数不改变实参的值 | | void swap2( int \*a, int b ); | x=2,y=2 | 不完全交换 | 传了x的地址，\*a=b;将b的值赋给了x，x的值被改变，但b形参不能改变y的值 | | void swap3( int \*a, int \*b ); | x=2,y=1 | 是 | 传入两个指针，通过指针改变了x,y的值 | | void swap4(int \*a, int \*b ); | x=1,y=2 | 否 | 交换的是指针，但没有改变&x,&y指向的值.  x,y并没变 |  1. 利用F10和F11功能键进行单步跟踪，4次运行观察各变量的变化情况，填写下表：  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **跟踪点(黄色箭头所指行)** | **实参x的值** | **实参y的值** | **跟踪点(黄色箭头所指行)** | **形参a(或\*a)的值** | **形参b(或\*b)的值** | | swap1调用行 | 1 | 2 | swap1函数左大括号处 | a的值： 1 | b的值： 2 | | swap1后的printf行 | 1 | 2 | swap1函数右大括号处 | a的值： 2 | b的值： 1 | | swap2调用行 | 1 | 2 | swap2函数左大括号处 | \*a的值： 1 | b的值： 2 | | swap2后的printf行 | 2 | 2 | swap2函数右大括号处 | \*a的值： 2 | b的值： 1 | | swap3调用行 | 1 | 2 | swap3函数左大括号处 | \*a的值： 1 | \*b的值： 2 | | swap3后的printf行 | 2 | 1 | swap3函数右大括号处 | \*a的值： 2 | \*b的值： 1 | | swap4调用行 | 1 | 2 | swap4函数左大括号处 | \*a的值： 1 | \*b的值： 2 | | swap4后的printf行 | 1 | 2 | swap4函数右大括号处 | \*a的值： 2 | \*b的值： 1 |   **实验题目（3）【见实验教材实验七的题目3】：**编写程序exp7\_3.c，从键盘读入一个字符串，去掉其中所有的空格得到一个新串后再输出（只能定义一个字符数组，不可以定义两个字符数组，可以根据编程需要定义字符指针以及其他变量）。  **实验解答：**   1. 写出完整的源程序代码并做适当注释：   #include <stdio.h>  #include <string.h>  int main()  {      char str[20];      char \*p,\*q;      printf("input a string:\n");      gets(str);      p = str;      q=str;      while (\*p != '\0')      {          if (\*p != ' ')          {              \*q = \*p;              p++;              q++;          }          else              p++;      }      \*q='\0';      printf("the new string is:\n");      puts(str);      printf("\n");      return 0;  }  ②按表中所给测试用例输入观察对应输出结果：   |  |  |  | | --- | --- | --- | | **测试用例要求** | **输入的原始串** | **输出结果串** | | 串中空格每处只有一个 | A bcd 12 45 t | Abcd1245t | | 至少有一处有连续多个空格字符 | A bc 12 t | Abc12t | | 字符串最前面是4个空格 | A b 12 t | Ab12t | | 你自己设计的测试用例 | abc de | abcde |   **四、实验小结**（包括问题和解决方法、心得体会、意见与建议、实验出错信息及解决方案等）  (一)实验中遇到的主要问题及解决方法  终端中文乱码，通过在settings.json中加入一段代码 "terminal.integrated.profiles.windows": {  "PowerShell": {  "source": "PowerShell", //一般推荐使用powershell  "overrideName": true,  "args": ["-NoExit", "/c", "chcp 65001"],  "icon": "terminal-powershell",  "env": {  "TEST\_VAR": "value"解决  实验7\_3中输入字符串时scanf不能读入空格，引用头文件string.h,使用gets函数可以读入空格  （二）实验心得  明白了传值和传址的区别，形参不能改变实参的值，学会通过指针改变实参的值  学会指针与数组结合，对数组进行排序，查找等操作  学会利用指针对字符串进行操作  （三）意见与建议（没有可省略）  **五、支撑毕业要求指标点** | | | | | | | | | | | | |
| **六、指导教师评语** | | | | | | | | | | | | |
| **评 分 细 则** | | **评分项** | | **优秀** | | **良好** | | **中等** | | **合格** | **不合格** | |
| **遵守实验室规章制度** | |  | |  | |  | |  |  | |
| **学习态度** | |  | |  | |  | |  |  | |
| **算法思想准备情况** | |  | |  | |  | |  |  | |
| **程序设计能力** | |  | |  | |  | |  |  | |
| **解决问题能力** | |  | |  | |  | |  |  | |
| **课题功能实现情况** | |  | |  | |  | |  |  | |
| **算法设计合理性** | |  | |  | |  | |  |  | |
| **算法效能评价** | |  | |  | |  | |  |  | |
| **回答问题准确度** | |  | |  | |  | |  |  | |
| **报告书写认真程度** | |  | |  | |  | |  |  | |
| **内容详实程度** | |  | |  | |  | |  |  | |
| **文字表达熟练程度** | |  | |  | |  | |  |  | |
| **其它评价意见** | |  | | | | | | | | |
| **本次实验能力达成评价（总成绩）** | |  | | **批阅人** | |  | | **日期** |  | |