

**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 | 实验二： 函数实验 | | | | | |
| 实验时间 | 2023 | 年 | 11 | 月 | 30 | 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** | | | **实验二 函数实验** | | | | | | **指导教师** | | 付竟芝 | |
| **实验类型** | | | **验证** | **实验学时** | | | **2** | | **实验时间** | | 11-30 | |
| 1. **实验目的和要求**   （1）掌握正确的函数定义与调用，需要时会正确使用函数声明。会正确设置形式参数，理解参数传递及程序的执行流程。  （2）理解各种不同存储类别变量的生命期与作用域。  （3）进一步熟悉调试器的使用，会利用调试器进行查错改错，会跟踪程序运行的每一步，观察变量的变化情况。 | | | | | | | | | | | | |
| 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）【见实验教材实验四的题目2】：**编程序exp4\_2.c，巧用函数调用，打印不同行数和字符构成的等腰三角形。具体要求：定义一个函数原型void DrawTriangle (int n,char c); ，实现功能为打印一个n行的由字符c组成的等腰三角形。主程序调用该函数，实现打印5行“ '**\*** ' ”、10行“ '**#** ' ”的等腰三角形。  **实验解答：**   1. 源程序exp4\_2.c的代码是：   #include <stdio.h>  //巧用函数调用，打印不同行数和字符构成的等腰三角形。  //具体要求：定义一个函数原型void DrawTriangle (int n,char c); ，  //实现功能为打印一个n行的由字符c组成的等腰三角形。主程序调用该函数，实现打印5行“ '\* ' ”、10行“ '# ' ”的等腰三角形。  void DrawTriangle(int n,char c)  {      int i,j;      for(i=1;i<=n;i++)      {          for(j=1;j<=n-i;j++)          {              printf(" ");          }          for(j=1;j<=2\*i-1;j++)          {              printf("%c",c);          }          printf("\n");      }  }  int main()  {      int n;      char c;      //printf("请输入行数：");      scanf("%d",&n);      //printf("请输入字符：");      scanf(" %c",&c);      DrawTriangle(n,c);      return 0;  } | | | | | | | | | | | | |
| ②主函数中只保留打印’\*’组成三角形的一条调用语句，然后改变控制行数的第一个实参，测试当该值为多大时，你的屏幕无法正常显示该等腰三角形，试分析原因。（提示：模拟DOS界面可以查看属性，观赛屏幕缓冲区的大小和窗口大小，再进行测试分析）。如果希望刚才的实参提供后仍然能正常显示，可以有什么办法？行数为86时无法正常显示，85正常。  我用的vs code，是在一个小终端里显示的，我使用二分法找到了测试值，  换用vs在黑框框中显示，情况有所不同，黑框框可以调大小  因为屏幕缓冲区大小和窗口大小不够。  可以缩放窗口，使之能显示更大的图形，更改窗口大小，右击窗口，属性—布局-窗口大小，窗口缓冲区大小，把数据调大即可  **实验题目（2）【见实验教材实验四的题目5】：**编程序exp4\_5.c，验证歌德巴赫猜想：2000以内的正偶数（不包括2）都能够分解为两个质数之和。（算法提示：将整数分解为两个整数，然后判断它们是否均为质数。若是，则满足题意并输出；否则重新进行分解和判断。其中，判断一个整数是否为质数采用函数实现。每个偶数只要得到一种分解就停止，不必要求出该偶数的所有质数和组合）。  **实验解答：**   1. 源程序exp4\_5.c的代码是：   #include <stdio.h>  int isPrime(int n)  {      int i;      for(i=2;i\*i<=n;i++)      {          if(n%i==0)              return 0;      }      return 1;  }  int main()  {      int i,j,k;      for(i=4;i<=2000;i+=2)      {          for(j=2;j<i;j++)          {              if(isPrime(j)&&isPrime(i-j))              {                  printf("%d=%d+%d\n",i,j,i-j);                  break;              }          }      }  }  实验题目（3）【见实验教材实验四的题目8】：用调试器观察程序exp4\_8.c的运行过程，并记录各种变量在每一步执行时的变化情况，在表格中填写每一个跟踪步每个变量对应的值。程序代码如下。  #include　<stdio.h>  int a = 1;  int f(int a)  {  auto int b = 2;  static int c = 3;  a = a+1;  b = b+1;  c = c+1;  return (a+b+c);  }  int main()  {  int i;  for (i=0;i<3;i++)  {  a = a+2;  printf("%d \n",f(a));  }  return 0;  }  **提示：**首先选择“调试”菜单，再选择“窗口”子菜单，再选择打开“监视1”窗口，在其中输入&a、&b、&c、&i进行观察，点开这些地址符前面的+号，就可以看到对应变量的值，注意观察地址值的变化以区分目前空间是哪个变量在作用域，尤其同名变量。下表中在某位置该变量不可见则不填内容。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **跟踪点（表示跟踪时的黄色箭头指在这一行）** | **全局**  **变量a** | **f的**  **形参a** | **f的自动**  **局部变量b** | **f的静态**  **局部变量c** | **主函数**  **的i** | | **main函数的左大括号处** | 1 |  |  |  |  | | **第1次for行** | 1 |  |  |  | 552（随机） | | a = a+2; | 1 |  |  |  | 0 | | printf行 | 3 |  |  |  | 0 | | **按F11到f函数左大括号处** | 1 |  | 0 | 3 |  | | auto int b = 2; |  | 3 | 7340152 | 3 |  | | a = a+1; |  | 3 | 2 | 3 |  | | b = b+1; |  | 4 | 2 | 3 |  | | c = c+1; |  | 4 | 3 | 3 |  | | return (a+b+c); |  | 4 | 3 | 4 |  | | **第1次f函数右大括号处** |  | 4 | 3 | 4 |  | | 按F10回到主函数的printf行 | 3 |  |  |  | 0 | | **按F10到第2次for行** | 3 |  |  |  | 0 | | a = a+2; | 3 |  |  |  | 1 | | printf行 | 5 |  |  |  | 1 | | **按F11到f函数左大括号处** | 1 |  | 1 | 4 |  | | auto int b = 2; |  | 5 | -8589993460 | 4 |  | | a = a+1; |  | 5 | 2 | 4 |  | | b = b+1; |  | 6 | 2 | 4 |  | | c = c+1; |  | 6 | 3 | 4 |  | | return (a+b+c); |  | 6 | 3 | 5 |  | | **第2次f函数右大括号处** |  | 6 | 3 | 5 |  | | 按F10回到主函数的printf行 | 5 |  |  |  | 1 | | **按F10到第3次for行** | 5 |  |  |  | 1 | | a = a+2; | 5 |  |  |  | 2 | | printf行 | 7 |  |  |  | 2 | | **按F11到f函数左大括号处** | 1 |  | 2 | 5 |  | | auto int b = 2; |  | 7 | -858993460 | 5 |  | | a = a+1; |  | 7 | 2 | 5 |  | | b = b+1; |  | 8 | 2 | 5 |  | | c = c+1; |  | 8 | 3 | 5 |  | | return (a+b+c); |  | 8 | 3 | 6 |  | | **第3次f函数右大括号处** |  | 8 | 3 | 6 |  | | 按F10回到主函数的printf行 | 7 |  |  |  | 2 | | **按F10到第4次for行** | 7 |  |  |  | 2 | | **主函数的return 0;处** | 7 |  |  |  | 3 |   **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  实验3  警告declaration of 'a' shadows a global declaration [-Wshadow]gcc  Gcc编译器报错说a的声明会遮蔽全局变量  说明同时存在全局变量a和自定义函数形参局部变量a时优先使用局部变量  实验一出现了中文乱码问题，本来想打印提示信息，请输入行数：结果到控制台上中文乱码了，只能不写提示语句了。  控制台大小不足导致无法完全显示图案，通过调大控制台大小解决  （二）实验心得  学会使用自定义函数解决问题，巧用函数调用  学会调试代码，在监视窗口观察变量的值，方便找到bug  （三）意见与建议（没有可省略） | | | | | | | | | | | | |
| 1. **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** | |  | |