

**网络安全与信息化学院**

**实验报告册**

2017 至 2018 学年 第 1 学期

**课 程**  高级语言程序设计

**专 业** 计算机类

**班 级** 计算机类174

**学 号** 171340139

**姓 名** 韩坤

**指导教师** 周碧英

**实 验 室** S420

网络安全与信息化学院制

二〇一七年

实验须知

1．进入实验室，应保持安静和整洁。

2．课前应认真预习本次实验内容及有关仪器设备介绍，基本了解实验基本原理，明确实验要求。

3．认真听取指导教师对仪器及设备的构造、基本原理、实验要求、注意事项等的讲解。

4．爱护仪器设备，细心操作，注意安全。实验中如发生意外或异常现象，应立即向指导教师报告。

5．实验结束后，应将所用的仪器及设备恢复为初始状态，收拾好桌凳，做好清洁工作。

6．实验报告是处理实验结果的总结资料，是考核学生实验成绩的主要依据之一，必须认真独立完成。课后一周内送交指导教师批阅，实验成绩按百分制计。

7．学生应规范填写实验报告，由实验指导教师给出填写样本。

8．本实验册应妥善保管，课程实验结束后应交回实验室。

本课程实验教学计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验课程名称 | 嵌入式程序设计 | | | |
| 实验课程类型 | □独立开课 | | ■附属于理论课 | |
| 实验总课时（大纲要求） | 32 | 实验项目个数（大纲要求必做） | | 9个 |
| 实验学分（独立开课的） |  | 适用专业 | 计算机类 | |
| 备注：该表由实验指导老师根据实验教学大纲指导学生统一填写。 | | | | |

本课程实验项目表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 实验项目名称（学生实验后填写） | 学时数 | 成绩 |
| 1 | 熟悉CodeBlock编程环境 | 2 |  |
| 2 | 结构化程序设计编程练习 | 4 |  |
| 3 | 函数应用编程 | 4 |  |
| 4 | 数组应用编程 | 4 |  |
| 5 | 综合练习 | 4 |  |
| 6 | 字符串的应用编程 | 2 |  |
| 7 | 指针的使用 | 4 |  |
| 8 | 结构体编程 | 2 |  |
| 9 | 文件编程 | 2 |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 备注；学生实验课程最后成绩由指导教师在课程实验结束后填写。 | | |  |

# 实验报告1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 熟悉CodeBlock编程环境 | | | | |
| 实验地点 | **S420** | 实验日期 | 2017.10.9/10.16 | 成绩 |  |
| 实验目的 | 熟悉CodeBlocks编程环境，掌握C语言程序执行过程 | | | | |
| 实验原理 | CodeBlock软件使用方法，C语言程序执行方法 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，Codeblock编程环境 | | | | |
| 实验内容与实验记录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| 实验要求：  1、熟悉环境，编写程序，求两个整数相加和相除的结果;  2、编写程序计算:1/2+2/3+3/4  3、编写程序验证符号常量：求圆的周长及面积。  4、编写程序计算：(2a+b)/(ab)  实验步骤：1、***第一步***：新建文件夹，保存文件，如图1-1所示：    图1-1 创建文件夹结果  ***第二步***：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序  ***第三步***：对问题进行分析，求两个整数之和，需要定义两个整数变量和一个存放和的变量,定义如下：  double a,b;  double s,d;  ***第四步***：两个整数需要用户从键盘输入，所以需要调用输入函数：scanf("%lf,%lf",&a,&b);  ***第五步***：当输入两个变量之后，就可以求和与商:s=a+b;d=a/b;  ***第六步***：按照题目要求输出两个之和：printf("所求两数之和为:%f,两数相除为：%f",s,d);  ***第七步***：完整的程序为：    ***第八步***：执行结果如图1-3所示:    图1-3  **程序二**：**编写程序计算**:1/2+2/3+3/4  ***第一步***：首先，要计算这个式子，需要一个存放结果的变量，由于式子已经确定，不需要定义其他的变量。定义如下：  double c;  ***第二步***：有了一个存放结果的变量后，我们就可以计算了，又因为计算机认为两个整数相除为整数，为了防止发生计算错误，所以用这样的方法来避免：  c=1./2+2./3+3./4;（即加上点，让计算机认为它是个小数）  ***第三步*：**算完后，为了更好地让用户知道他算的是什么，所以我们用printf函数来告诉用户他所求的东西是什么以及结果，如下：  printf("(1/2)+(2/3)+(3/4)=%f",c);  ***第四步***:完整的程序为:    ***第五步***:执行结果如图2-1所示:    2-1  **程序三:编写程序验证符号常量：求圆的周长及面积**  ***第一步***:因为C语言不能识别π，所以我们要使用define宏观定义一个π。即:  #define pl 3.14  ***第二步***：现在我们需要引入三个变量r,c,s，r表示圆的半径，c表示圆的周长，s表示圆的面积.即:  double r;  double c,s;  ***第三步***：接下来我们要让用户输入半径，用给予文字提示，用printf函数，又引入半径需要由键盘输入，所以再引入一个scanf函数.即:  printf("请输入圆形的半径");  scanf("%lf",&r);  ***第四步***：现在进行计算  c=2\*r\*pl;  s=pl\*r\*r;  ***第五步***：输出结果，并给予文字提示:  printf("圆形的周长和面积分别是:%f,%f",c,s);  ***第六步***:完整的程序为:    ***第七步***：执行的结果为(如图3-1所示)：    3-1  **程序四: 编写程序计算**：(2a+b)/(ab)  ***第一步***:首先需要定义3个变量a,b,c,用c表示(2a+b)/(ab)的值.即:  double a ,b;  double c;  ***第二步:***让用户输入两个数,a,b,并用scanf函数提供占位符,即  printf("请输入两个数:\n");  scanf("%lf,%lf",&a,&b);  第三步:进行计算,如下  c=(2\*a+b)/(a\*b);  ***第四步***:为了让用户更直观的看出所求的是什么,所以表示为  printf("(2a+b)/(ab)=%f",c);  ***第五步:*** 完整的程序为:    ***第六步***:运行结果如图4-1所示    4-1 | | | | | |
| 实验结果分析及结论 | | | | | |
| 分析:#include<stdio.h>为预处理命令;  Int为整形函数,double为双精度函数;  Printf为输出指令;  Scanf为输出指令;  \n为换行;  结论:如果程序中需要用户输入数字或其他,那么需要先给这个数字进行定义,有int,float,double.一般来说,使用printf函数进行输出时,也应该有scanf函数对应输入,并且要注意必须给一些文字提示. | | | | | |

# 实验报告2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 数据处理编程练习 | | | | |
| 实验地点 | **519** | 实验日期 |  | 成绩 |  |
| 实验目的 | 通过程序熟悉C语言中各种数据类型及运算符的使用 | | | | |
| 实验原理 | C语言数据类型的定义及使用方法、运算符的优先级和结合方向 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验内容：**  1、验证课本第3章中的例题  2、编写程序，用于计算圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。圆半径、圆柱高由用户通过键盘输入。要求输入输出有相应的文字说明，输出结果保留两位小数。  3、编写程序，输出一个四位数各个位上数字之和及其乘积，原始整数由键盘输入。例如键盘输入为1234，则输出结果应该为10和24。  4、编写程序，从键盘上输入一个浮点型数，保留小数两位输出，并对小数第三位进行四舍五入处理，要求不得使用格式修饰符进行输出处理。  5、编写程序，将输入的两个整数number1和number2合并成一个新的整数number，number1的十位是number的百位，number1的个位是number的千位；number2的十位是number的个数，number2的个数是number的十位。  6、编写程序验证转义字符常量：例2-1、输出“abc\rd”、“abc\bde”。  **实验步骤：** 第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序  程序一: 编写程序，用于计算圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。圆半径、圆柱高由用户通过键盘输入。要求输入输出有相应的文字说明，输出结果保留两位小数。  第一步:D首先定义几个变量，分别存放题目中所需要的，代码如下：  double r,h;  double yzc,ymj,qbmj,qtj,ztj;(因为，圆面积等有可能为小数，所以用double定义)  第二步：让用户输入圆半径、圆柱高，并给予文字提示：  printf("请输入圆半径与圆柱高:\n");  scanf("%lf,%lf",&r,&h);  第三步:对所求的进行计算：  yzc=2\*pl\*r;  ymj=pl\*r\*r;  qbmj=4\*r\*r\*pl;  qtj=4./3\*pl\*r\*r\*r;  ztj=yzc\*h;  第四步：输出结果：  printf("圆周长为:%.2f\n圆面积为:%.2f\n圆球表面积为:%.2f\n圆球体积为:%.2f\n圆柱体积为:%.2f\n",yzc,ymj,qbmj,qtj,ztj);  第五步：完整程序为：    第六步：运行结果为：    程序二: 编写程序，输出一个四位数各个位上数字之和及其乘积，原始整数由键盘输入。例如键盘输入为1234，则输出结果应该为10和24。  第一步:定义一个表量来表示这个四位数,:  int a;  第二步:让用户输入一个四位数:  printf("请输入一个四位整数:");  scanf("%d",&a);  第三步:再定义4个变量,来存放这个四位数的个十百千位:  unsigned num1,num2,num3,num4;  第五步:讲这个四位数上的数字分离:  num1=a%10; //分离个位数字  num2=a/10%10; //分离十位数字  num3=a/100%10; //分离百位数字  num4=a/1000%10; //分离千位数字  第六步:输出所要求的:  printf("%d\n",num1+num2+num3+num4);  printf("%d\n",num1\*num2\*num3\*num4);  第七步:完整程序为:    第八步:运行结果为:    程序三: 编写程序，从键盘上输入一个浮点型数，保留小数两位输出，并对小数第三位进行四舍五入处理，要求不得使用格式修饰符进行输出处理。  第一步:定义两个变量,用于存放,浮点型数和保留两位小数并四舍五入后的:  double a,b;  第二步:让用户输入一个浮点型数:  printf("请输入一个浮点型数：");  scanf("%lf",&a);  第三步:对这个浮点型数进行处理并输出:  b=((int)(a\*100+0.5))\*0.01;  printf("结果为：%f\n",b);  因为要给这个数保留两位小数,并且要四舍五入,所以先给用户输入的这个浮点型数乘100再加上0.5,如果原先的浮点型数的小数点后第三位是大于等于5的,那么加上0.5后则进一位,如果小于5则不变,然后将这个数强制取整,最后再乘0.01,得到的就是最终想要的结果  第四步:完整程序为:    第五步:运行的结果为:    程序四: 编写程序，将输入的两个整数number1和number2合并成一个新的整数number，number1的十位是number的百位，number1的个位是number的千位；number2的十位是number的个数，number2的个数是number的十位。  第一步:定义3个变量,分别表示用户输入的两个数以及最后重新整合后的数  int u\_number1,u\_number2;  int u\_number;  第二步:由用户输入这两个数:  printf("请输入两个两位数:\n");  scanf("%d,%d",&u\_number1,&u\_number2);  第三步:定义4个数来表示输入的2个整数的十位和个位:  unsigned num4,num5,num6,num7;  num4=u\_number1%10;//分离第一个数字的个位  num5=u\_number1/10%10;//分离第一个数字的十位  num6=u\_number2%10;//分离第二个数字的个位  num7=u\_number2/10%10;//分离第二个数字的十位  第四步:整合成所要求的数并输出:  u\_number=num4\*1000+num5\*100+num6\*10+num7;  printf("%d\n",u\_number);  第五步:完整程序为:    第六步:运行结果为:    程序五: 编写程序验证转义字符常量：例2-1、输出“abc\rd”、“abc\bde”。  ***第一步:***用printf函数运行abc/rd和abc\bde:  printf("abc\rd\n");  printf("abc\bde\n");  第二步:完整代码为:    第三步:运行结果为: | | | | | |
| 实验结果分析及结论  55555 | | | | | |
| **1:**#define pl 3.14这种为宏定义,定义Pl为3.14  **2:**在最后一个程序中,\r的作用为回到本行开头,所以当程序运行到abc时碰到了\r回到开头后再输出d,所以结果为dbc;\b的作用为退格回到本行开头所以运行结果为:abde.  **3:**%10可以取一个整数的个位.  **4:**要取出一个数的每个位的数字时,可以采用%10的方法,并且在适当时候使用强制转换来得到自己需要的数. | | | | | |

# 实验报告3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 选择结构程序设计 | | | | |
| 实验地点 | **519** | 实验日期 |  | 成绩 |  |
| 实验目的 | 通过程序熟悉C语言关系运算符、逻辑运算符和条1:算符的用法、选择结构的应用。 | | | | |
| 实验原理 | C语言选择结构思想 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  1、验证课本第4章中的例题  2、编写程序，从用户输入的4个整数中找出最大数和最小数。要求有明确的交互界面。  3、用Switch语句编写程序实现对分数分等级输出。要求有明确的交互界面。  4、编写程序，要求输入整数，a和b，如果a-b为奇数，则输出a和b个位数字的乘积；否则输出a和b个位数字的和。  5、编写程序输出三角形的面积和周长，边长由用户通过键盘输入，输入时应该考虑输入的三个数是否能构成一个三角形。  6、用Switch语句实现计算器的四则运算。  **实验步骤：**   第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一: 编写程序，从用户输入的4个整数中找出最大数和最小数。要求有明确的交互界面。 第一步:定义6个变量,用于存放这四个数和其中的最大最小值:  int a,b,c,d;  int max,min;  第二步:让用户输入这4个数:  printf("Enter four integer:");  scanf("%d,%d,%d,%d",&a,&b,&c,&d);  第三步:比较大小,先求出最大最小值  if(a>b)max=a;//如果a大于b，则将a赋给max  else max=b;//在a,b中选出最大值  if(c>max) max=c;  if(d>max) max=d;//用max再和与c,d做比较  同理：最小值：  if(a<b) min=a;  else min=b;  if(c<min) min=c;  if(d<min) min=d;//原理同上  第四步：将最大最小值输出：  printf("Largest:%d\nSmallest:%d\n",max,min);  第五步：完整程序为：    第六步：运行结果为：   程序二: 用Switch语句编写程序实现对分数分等级输出。要求有明确的交互界面。 第一步:定义一个int型的变量来存放成绩,并让用户输入:  int a;  printf("请输入成绩：\n");  scanf("%d",&a);  第二步:用switch…case语句对成绩分等级:  switch(a)  {  case 90 ... 100://成绩为90到100  printf("A");//则显示A  break;//退出语句  case 80 ... 89:  printf("B");  break;  case 70 ... 79:  printf(" C");  break;  case 60 ... 69:  printf(" D");  break;  case 0 ... 59:  printf("F");  break;  default:  printf("输入错误！\n");//不在0到100的则显示输入错误  break;  第三步:完整程序为:      第四步:运行结果为:    程序三: 编写程序，要求输入整数，a和b，如果a-b为奇数，则输出a和b个位数字的乘积；否则输出a和b个位数字的和。  第一步:定义四个变量,用于存放这两个整数a、b以及这两个整数的个位：  int a,b;  int x,y;  第二步：让用户输入这两个整数：  printf("请输入两个整数:");  scanf("%d,%d",&a,&b);  第三步：取出这两个整数的个位：  x=a%10;//取出a的个位  y=b%10;//取出b的个位  第四步：对a-b进行判断：  ((a-b)%2)!=0?printf("a,b个位数字乘积为：%d\n",x\*y):printf("a,b个位数字和为：%d\n",x+y); //如果a-b是奇数，则输出这两个数个位的乘积，如果为偶数，则输出这两个数个位的和  第五步：完整程序为：   第六步：运行结果为：  程序四: 编写程序输出三角形的面积和周长，边长由用户通过键盘输入，输入时应该考虑输入的三个数是否能构成一个三角形。 第一步:定义5个变量,来存放三角形的三边以及周长和面积:  double a,b,c,z,s,area;  第二步:让用户输入三角形的三条边:  printf("请输入三角形的三条边:");  scanf("%lf,%lf,%lf",&a,&b,&c);  第三步:计算面积和周长:  如果输入的可以构成三角形则就行计算,否则提示输入数据不能构成三角形:  if(a+b>c&&a+c>b&&b+c>a&&a\*b\*c>0) //判断是否能构成三角形  {  s=0.5\*(a+b+c);  area=sqrt(s\*(s-a)\*(s-b)\*(s-c)); //计算面积  z=a+b+c; //计算周长  printf("面积为%f,周长为%f",area,z);  }  else  printf("输入的数据不能构成三角形");  第四步:完整的程序为:    第五步:运行结果为:   程序五: 用Switch语句实现计算器的四则运算 第一步:定义2个数以及一个字符型变量,字符型变量来存放运算符:  int a,b;  char c;  第二步:让用户进行输入:  printf("请输入两个数：");  scanf("%d%c%d",&a,&c,&b);  第三步:用switch…case来选择用户输入的运算符:  Switch(c)  {  case'+':  printf("%d+%d=%d\n",a,b,a+b);  break;  case'-':  printf("%d-%d=%d\n",a,b,a-b);  break;  case'\*':  printf("%d\*%d=%d\n",a,b,a\*b);  break;  case'/':  printf("%d/%d=%d\n",a,b,a/b);  break;  default:  printf("输入错误！\n");  break;  }  第四步:完整的程序为:    第五步:运行结果: | | | | | |
| 实验结果分析及结论 | | | | | |
| **1:在程序中,可以使用if-else来对程序进行一些限制,以实现自己的目的.**  **2:a?b:c :如果满足a则执行b否则执行c,利用此语句可以将一些简单的if-else语句用这个表示**  **3:对于一些字符型的输入输出,应该用%c输入输出,用char进行定义**  **易错点:**  **1：没有注意运算符的先后顺序**  **2：解题过程中出现逻辑错误** | | | | | |

# 实验报告4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 循环结构程序设计 | | | | |
| 实验地点 | **519** | 实验日期 |  | 成绩 |  |
| 实验目的 | 通过程序熟悉C语言结构化程序设计思想 | | | | |
| 实验原理 | C语言顺序结构、选择结构和循环结构思想 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  1、验证课本第5章中的例题  2、输入一个字符串，对其中字母、数字和其他符号分别计数并输出。  3、编写程序，计算正整数1到n之间的奇数之和和偶数之和。  4、编写程序计算一个4位正整数n的各位数字之和。例如输入数字为1234，则输出应该为1+2+3+4=10；  5、输出100-1000之内所有的素数，每行显示10个。  6、编写程序，输入一个自然数存放在k中，然后在1到k的自然数中，统计各位数组之和为5的奇数个数。  **实验步骤：** 第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一: 输入一个字符串，对其中字母、数字和其他符号分别计数并输出。 第一步:定义一个字符型变量,再定义4个无符号的整型变量,用于存放数字、字母、其他符号、总和的个数:  char chars;  unsigned a,b,c,d;  第二步：令a、b、c、d初值为0：  a=0,b=0,c=0,d=0;  第三步：提示用户输入一串字符  printf("请输入一串字符，以#结束：\n");  第四步：利用while循环,对所输入的字符串一个一个进行判断:  while((chars=getchar())!=‘#')//当输入的字符不是#时就行循环  {  if(chars>=‘A'&&chars<=‘Z'||chars>=‘a'&&chars<=‘z')//判断字符是否为字母  a++;//如果是，a加1  else if(chars>=‘0'&&chars<=‘9')//如果为数字的话  b++;//b加1  else//其余的则为符号  d++;//d加1  c++;//这是总共输入的字符  }  第五步：对所统计的进行输出：  printf("数字有：%d",b);  printf("字母有：%d",a);  printf("其他有：%d",d);  printf("总字符有：%d",c);  第六步：完整程序为：    第七步：运行结果为：   程序二：编写程序，计算正整数1到n之间的奇数之和和偶数之和。 第一步：定义4个变量，用于存放n，奇数和，偶数和，循环初值i：  int n,sum1,sum2,i;  第二步：令sum1,sum2的初值为0：  sum1=0;sum2=0;  第三步：让用户输入n并对n进行限制必须大于一，否则重新输入：  do{  printf("计算正整数1到多少的和：");  scanf("%d",&n);  }while(n<1);  第四步：利用for循环求出所需要的值：  for(i=1;i<=n;i++) //循环n次  {  if(i%2==0)//当i是偶数时  sum1+=i;// 加到sum1上  else//为奇数时  sum2+=i;//加到sum2上  }  第五步：输出奇数和、偶数和：  printf("偶数之和为：%d",sum1);  printf("奇数之和为：%d",sum2);  第六步：完整程序为：    第七步：运行结果为：   程序三：编写程序计算一个4位正整数n的各位数字之和。例如输入数字为1234，则输出应该为1+2+3+4=10； 第一步:定义4个变量用于存放输正整数、循环初值、每个位相加的结果、以及产生的每个新数的个位,并定义sun=0;  int n,i,sum,n1;  sum=0;  第二步：让用户输入n，并进行限制：  do{  printf("请输入一个正整数：");  scanf("%d",&n);  }while(n<0);//限制输入，必须输入大于0  第三步：利用for循环来解决：  for(i=0;;i++)//循环条件，为无限循环  {  n1=n%10;//对这个数取余  n/=10;//除10取整  sum+=n1;//将所得到的每位数相加  if(n==0)//如果n等于0了，则结束循环  break;  }  第四步：输出结果：  printf("%d\n",sum);  第五步：完整程序为：    第六步：运行结果为：   程序四：输出100-1000之内所有的素数，每行显示10个 第一步：定义3个整型变量，2个位循环初值，另一个为计数器：  int k;  int j;  int i=0;  第二步：对100到1000内的素数进行输出：  for(j=100;j<=1000;j++)//从100开始验证，到1000为止  {  for(k=2;k<j;k++)//让这个数从2开始除，直到它本身  if(j%k==0)//如果这个数碰到了一个数，除它取余为0  break;  if(k==j)//若这个数和它是一样的，则为素数。  {  printf("%5d ",j);//输出这些数  i++;//用i来做计数器  if(i%10==0)//每行留10个  printf("\n");//然后换行  }  第三步：完整程序为：    第四步：运行结果为：   程序五：编写程序，输入一个自然数存放在k中，然后在1到k的自然数中，统计各位数组之和为5的奇数个数。 第一步：定义所需要的变量：  int n,i,j,n1,sum,k;  k=0;  第二步：让用户输入k，并进行限制：  do{  printf("请输入一个数（>1）");  scanf("%d",&n);  }while(n<=1);  第三步：利用循环求解：  for(i=1;i<=n;i++)//循环k次  {  j=i;//将i的值赋给j  sum=0;//令sun在每一次循环里初值都是0  while(1)  {  n1=j%10;//用n1存放j的个位  sum+=n1;//将这些数相加  j/=10; //j除10取整  if(j==0)//当k等于0时，退出while循环，否则重复，这样可以将这个数的每个位的数字相加  break;  }  if(sum==5&&i%2!=0)//如果此时的i是奇数且各个位数和为5  k++; //则n加一  }  第四步：完整程序为：    第五步：运行结果为： | | | | | |
| 实验结果分析及结论 | | | | | |
| **1：对一个数的每个位进行求和时，可以采用for循环来求**  **2：如果不知道循环多少次，则可以先设定为无限循环，在循环语句里再增加判断条件**  **易错分析：**  **1：在循环中，会不注意循环初值的变化而导致错误**  **2：不能够清楚知道哪些语句是在循环中的，哪些在循环外**  **3：逻辑错误造成循环体不循环或者无限循环**  **4：如果进行累加或者累乘，应该给给存储变量赋予初值** | | | | | |

# 实验报告5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 函数应用编程 | | | | |
| 实验地点 | **519** | 实验日期 |  | 成绩 |  |
| 实验目的 | 熟练掌握函数的定义及其调用方法 | | | | |
| 实验原理 | 函数的嵌套调用和函数的递归调用 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  1、验证课本第6章中的例题  2、编程序，采用递归调用方法实现将一个长整数从右向左按位输出。  3、编写程序验证，一个大偶数可分解为两个素数之和。  4、编写程序，求s=1！+2！+3！+4！+…+n！。  5、用两种方法输出斐波那契数列：  第一种：循环；第二种：递归调用  6、用两种求两个数的最大公约数  第一种：循环；第二种：递归调用  **实验步骤：** 第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一：编程序，采用递归调用方法实现将一个长整数从右向左按位输出。 第一步：定义一个函数用于将输入的整数逆顺序输出：  void ni(int n)//定义一个无返回值的函数  {  printf("%d",n%10); //输出这个数的个位  if(n/10!=0) //如果这个数除10取整不等于0时  ni(n/10); //递归n/10取整,  }  第二步:在main函数里定义一个n,然后让用户输入n,并对这个数进行限制:  int n;  do{  printf("请输入一个整数：");  scanf("%d",&n);  }while(n<0);  第三步:提示用户逆向输出的数为:  printf("%i的逆顺序输出为：",n);  第四步:调用逆向输出的函数:  ni(n);  printf("\n");  第五步:完整程序为:    第六步:运行结果为:   程序二: 编写程序验证，一个大偶数可分解为两个素数之和。 第一步:定义一个判断是否为素数的函数:  int sushu(int n)  {  int i;  for(i=2;i<=n/2;i++)  if(n%i==0)  return 0;//如果除i取余为0时,返回0  return 1;//反之,返回1  }  第二步:定义一个数,让用户输入一个偶数,并进行限制:  int even\_n;  do{  printf("请输入一个偶数:");  scanf("%d",&even\_n);  }while(even\_n<=0||even\_n%2!=0);  第三步:定义一个数,让这个数从二开始循环,逐渐加1  int n;  for(n=2;n<even\_n;n++)  {  if(sushu(n)&&sushu(even\_n-n))//如果n和输入的这个偶数减去n都为素数,则输出  {printf("%d可分解为%d+%d\n",even\_n,n,even\_n-n);  break;}  }  if(n>even\_n)//如果n大于了输入的偶数,则证明失败  printf("结论错误,%d不能分解为两个素数之和!\n",even\_n);  第四步:完整程序为:    第五步:运行结果为: 程序三: 编写程序，求s=1！+2！+3！+4！+…+n！ 第一步:定义一个返回阶乘的函数:  double jiecheng(int n)//返回n的阶乘  {  double p=1;//定义p初值为1  int i;  for(i=1;i<=n;i++)  p\*=i;//循环求n的阶乘  return p;//返回p(即n的阶乘)  }  第二步:定义两个数:n,i :  int n,i;  第三步:让用户输入要求1到多少的阶乘和:  do{  printf("求1到几的阶乘和（>1):");  scanf("%d",&n);  }while(n<=1);  第四步:定义一个num,初值为0,然后循环求阶乘和:  double sum=0;  for(i=1;i<=n;i++)  sum+=jiecheng(i);//将函数jiecheng的值加起来  printf("和为：%f",sum);  第五步:完整程序为:    第六步:运行结果为:   程序四: 用两种方法输出斐波那契数列：第一种：循环 第一步:定义f1,f2都为1,再定义两个变量:  int n,f1=1,f2=1,i;  第二步:让用户输入要求数列的前多少项:  printf("要求数列的前多少项：");  scanf("%d",&n);  第三步:使用循环:  for(i=1;i<=n/2;i++)//循环到n/2  {  printf("%5d%5d",f1,f2);//先输出f1,f2  f1=f1+f2;  f2=f1+f2;//计算数列  }  第四步：考虑当n为奇数时的情况：  if(n%2!=0)//如果输入的n是奇数的话  printf("%5d",f1);//则多输出一个f1  第五步：完整程序为：    第六步：运行结果为：   第二种：递归调用 第一步：定义一个返回数列值的函数：  int shulie(int n)  {  if(n==1||n==2)  return 1;//如果n为1或2时,返回1  else  return shulie(n-1)+shulie(n-2);//其余的则递归返回  }  第二步：让用户输入要求数列的前多少项：  printf("要求数列的前多少项:");  scanf("%d",&n);  第三步：用循环输出数列：  for(i=1;i<=n;i++)  {  printf("%15d",shulie(i));//从第一项开始输出,直到第n项  j++;  if(j%5==0)每五个换行  printf("\n");//换行  }  第四步：完整程序为：    第五步：运行结果为：   程序五：用两种求两个数的最大公约数第一种：循环 第一步：让用户输入两个数，并对其进行限制：  int m,n,i;  do{  printf("请输入两个正整数，先小后大：");  scanf("%d,%d",&m,&n);  }while(m\*n<=0||m>n);  第二步：使用循环求最大公约数：  for(i=m;i>0;i--)//从用户输入的小值开始递减  {  if(m%i==0&&n%i==0)//当输入的这两个数除i都为时，  {  printf("%d",i);//此时i为最大公约数  break;}//然后退出  }  第三步：完整程序为：    第四步：运行结果为：   第二种:递归调用 第一步:定义一个函数开判断最大公约数:  void yue(int m,int n,int i)  {  if(m%i==0&&n%i==0)  printf("%d",i);//当m,n除i都余0时,  else  yue(m,n,i-1);//要不然,返回m,n,i-1  }  第二步:让用户输入两个数，并对其进行限制：  int m,n,i;  do{  printf("请输入两个正整数，先小后大：");  scanf("%d,%d",&m,&n);  }while(m\*n<=0||m>n);  第三步:调用函数:  yue(m,n,m);//返回的第三个m为函数中的I,这是为了防止递归时改变m的值  第四步:完整程序为:    第五步:运行结果为: | | | | | |

|  |
| --- |
| 实验结果分析及结论 |
| 易错点:  1:在使用函数递归时,要注意防止改变了某一值而影响效果  2:定义函数时不能使用int m,n,只能使用int m,int n这种格式  3:如果要先声明函数,则必须加分号,如果在main函数前直接定义,则不用 |

# 实验报告6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 数组应用编程 | | | | |
| 实验地点 | **519** | 实验日期 |  | 成绩 |  |
| 实验目的 | 熟练掌握数组的应用 | | | | |
| 实验原理 | 一维数组、二维数组 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  1、编写程序实现，将十进制数转换为R（2-16）进制数。  2、编写程序实现，在一个有序序列中插入一个数，使序列仍然有序。  3、编写程序，求二维数组中每行的最大值。  4、编写程序，在二维数组中找一个数，这个数满足在它所在行上最大， 同时在它所在列上最小。  5、输入一个一维数组，然后对一维数组逆序存放。  **实验步骤：** 第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一：编写程序实现，将十进制数转换为R（2-16）进制数。 第一步：定义所需要的数和数组：  char ch[32];  int n,r,n1;  int i=0;//定义i的初值为0  第二步：让用户输入要转换的数和要转换成多少进制：  do{  printf("请输入一个正整数：");  scanf("%d",&n);  }while(n<=0);//让用户输入一个十进制整数  do{  printf("要转换为几进制数：");  scanf("%d",&r);  }while(!(r>=2&&r<=16));//让用户输入想要把这个数转化为几进制的数  第三步：讲这个数赋给n1；  n1=n;//将这个值赋给n1  第四步：进行循环  while(n1)//当n1不等于0时循环  {  if(n1%r>=10)  ch[i]=n1%r+55;//如果这个数除r取余的余数大于等于10，则将这个余数加55转化为字母  else  ch[i]=n1%r+48;//反之，转换为数字字符  n1/=r;//除r取整  i++;//i加1  }  第五步：对结果进行逆序输出  printf("%d的%d进制为：\n",n,r);  while(--i>=0)//逆顺序输出，当小于0时停止循环  {  printf("%c",ch[i]);//输出这些值  }  第六步：完整程序为：    第七步：运行结果为：   程序二：编写程序实现，在一个有序序列中插入一个数，使序列仍然有序。第一种：输入时有序： #include<stdio.h>  int main()  {    int n;  do{  printf("输入原始序列的个数：");  scanf("%d",&n);  }while(n<=1); //让用户输入他要输入的原始序列的个数  int a[n]; //定义一个数组，里面有n个数  int i=0； //定义一个变量i，初值为0  printf("输入%d个原始数列：\n",n); //提示用户让他输入n个原始数列    printf("输入数列中的第%d个数：",i+1);  scanf("%d",&a[i]); //让用户先输入原始数列的第一个数  do{  printf("输入数列中的第%d个数：",i+2); //给予用户提示  scanf("%d",&a[i+1]); //用循环让用户输入接下来的数  if(a[i]>a[i+1])  {  printf("顺序错误，输入的数应不小于%d\n",a[i]);//如果输入错误，提示用户不能小于前一个数  continue; //如果用户没有按从小到大的顺序输入，则返回重新输入  }  i++; //i加1  }while(i<n-1);  printf("原始数列为：\n");  for(i=0;i<n;i++)  printf("%5d",a[i]);  printf("\n"); //将原始序列输出  char ch; //定义一个字符型的变量  do{  int num;  do{  printf("请输入要插入的数：");  scanf("%d",&num);  }while(num<0); //让用户输入要插入的数并对其进行限制  int i,j; //定义两个变量用于循环  for(i=0;i<n;i++)  if(num<=a[i])  break;// 让要插入的数一一与原始序列中的比较，直到要插入的这个数小于原始序列中某个数时，退出  if(i==n)//如果此时i=n的话，即要插入的这个数在这个序列里最大  a[i]=num; //则把这个数赋给a[i]  else  {  for(j=n-1;j>=i;j--)  a[j+1]=a[j]; //如果是其他情况，则把这个数后的的序列全部向后推一个数  a[i]=num; //然后再把要插入的这个数赋给a[i]  }  n++;  printf("插入后的数据序列为：");  for(i=0;i<n;i++)  printf("%10d",a[i]);  printf("\n");//将插入后的数据序列输出  printf("继续(Y/N)？:");//询问用户是否继续  while(getchar()!='\n')continue;//清除缓冲区多余字符  ch=getchar();//输入  }while(ch=='Y'||ch==’y’); //用户选择Y或者y后继续插入  }  **运行结果为：**   第二种：输入的时候不进行排序控制，利用排序数组排序 #include<stdio.h>  void shuru(int a[],int n)//定义输入函数  {  printf("请输入%d个数（用空格隔开）:\n",n);  int i;  for(i=0;i<n;i++)  scanf("%d",&a[i]);//让用户输入n个数，用空格隔开  }  void paixu(int a[],int n)//定义排序函数  {  int i,t,j;  for(j=0;j<n;j++)  for(i=0;i<n-1;i++)  if(a[i]>a[i+1])  {  t=a[i];  a[i]=a[i+1];  a[i+1]=t;//如果前面的数大于后面的数，则交换位置  }  }  void shuchu(int a[],int n)  {  int i;  printf("原始数列为：\n");  for(i=0;i<n;i++)  printf("%5d",a[i]);  printf("\n");//输出排序后的原始序列  }  int main()  {  int n,i;  do{  printf("输入原始序列的个数:");  scanf("%d",&n);  }while(n<1); //输入原始数据的个数  int a[n]; //定义一个数组  shuru(a,n); //定义一个输入函数,对这些数进行输入  paixu(a,n); //对这些数进行排序  shuchu(a,n); //输出  char ch;  do{  int num;  printf("请输入要插入的数：");  scanf("%d",&num);//选择要插入的数  for(i=n-1;i>=0;i--)//从原始序列最后一个数开始比较  {  if(num<=a[i])//如果这个数小于a[i]的话  {  a[i+1]=a[i];a[i]=num;//将a[i]推后一个数，将num赋给a[i]  }  else if(num>a[n-1])  a[n]=num; //如果要插入的这个数大于数组里最后一个元素，则直接将要插入的这个数放在数组的最后，即a[n]  else  break;  }  n++;//n加1  printf("插入后的数据序列为：");  for(i=0;i<n;i++)  printf("%10d",a[i]);//将插入后的数据序列输出  printf("\n");  printf("继续(Y/N)？:");//询问用户是否继续  while(getchar()!='\n')continue;//清除缓冲区多余字符  ch=getchar();  }while(ch=='Y'||ch=='y');//用户选择Y或者y后继续插入  }  **运行结果为：**   程序三：编写程序，求二维数组中每行的最大值。 #include<stdio.h>  int main()  {  int i,j,max,m,n;  do{  printf("输入数组的行和列(用逗号隔开):");//让用户输入数组的行和列  scanf("%d,%d",&m,&n);  }while(m<1||n<1);  int a[m][n];//定义一个m行n列的二维数组  for(i=0;i<m;i++)  {  printf("请输入第%d行的%d个数:",i+1,n);//对需要输入的进行提醒  for(j=0;j<n;j++)  scanf("%d",&a[i][j]);//让用户输入数组中的数  }  printf("矩阵为:\n");  for(i=0;i<m;i++)  {  for(j=0;j<n;j++)  printf("%5d",a[i][j]);  printf("\n");  }//对赋值后的矩阵进行输出  for(i=0;i<m;i++)  {  max=a[i][0];//将这一行第一个数付给max作为处置  for(j=0;j<n-1;j++)  {  if(max<a[i][j+1])//如果max后面一个数大于max，  max=a[i][j+1];//则将这个数赋给max  }  printf("第%d行的最大值为:%d\n",i+1,max);//然后输出第i+1行的最大值  }  }  **运行结果为：**   程序四：编写程序，在二维数组中找一个数，这个数满足在它所在行上最大， 同时在它所在列上最小。 #include<stdio.h>  int main()  {  int i, j, max, m, n, min, c;  int k;  do {  printf("输入数组的行和列(用逗号隔开):");  scanf("%d,%d", &m, &n);  } while (m<1 || n<1);//让用户输入数组的行和列，并对其输入进行限制  int a[m][n];//定义一个二维数组  for (i = 0; i<m; i++)  {  printf("请输入第%d行的%d个数:", i + 1, n);  for (j = 0; j<n; j++)  scanf("%d", &a[i][j]);  }//让用户输入二维数组的元素  printf("矩阵为:\n");  for (i = 0; i<m; i++)  {  for (j = 0; j<n; j++)  printf("%5d", a[i][j]);  printf("\n");  }//对其矩阵进行输出  for (i = 0; i<m; i++)  {  max = a[i][0];//让行最大max为这一行的第一个元素  for (j = 1; j<n; j++)  if (max <= a[i][j])  {  max = a[i][j];//遍历这一行，如果有元素比max大，则将这个元素赋给max  k = j;//将此时的列数j赋给k  min = max;//将最大值赋给min  }  for (c = 0; c<m; c++)  if (min>a[c][k])  {  min = a[c][k];//遍历最大值所在的这一列，  }  if (min == max)//如果此时max=min，则说明这行的行最大在这一列是最小的  printf("%d在第%d行上最大,并且在第%d列上最小\n", max, i+1, k + 1);  }  }  **运行结果为：**   程序五：输入一个一维数组，然后对一维数组逆序存放。 #include<stdio.h>  int main()  {  int n,i;  do{  printf("请输入数组元素的个数：");  scanf("%d",&n);  }while(n<=0);//让用户输入数组个数  int a[n];//定义个一维数组  printf("请输入%d个数:",n);  for(i=0;i<n;i++)  scanf("%d",&a[i]);//让用户输入这些数  printf("数组逆顺序存放后为：");  for(i=n-1;i>=0;i--)  printf("%5d",a[i]);//从数组最后一个元素开始输出，实现逆顺序存放  }  **运行结果为：** | | | | | |
| **实验结果分析及结论** | | | | | |
| **1：分析程序时给一些变量赋予初值，避免发生错误。**  **2：数组的下标最小为0，而不是1。**  **3：二维数组的每输出完一行应该换行，形成矩阵样式**  **错误：**  **1：将循环次数搞错** | | | | | |

# 实验报告7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 指针及字符串的使用 | | | | |
| 实验地点 | **519** | 实验日期 | 2017.12.25 | 成绩 |  |
| 实验目的 | 熟练掌握指针的用法 | | | | |
| 实验原理 | 指针的定义，初始化 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  一、验证：指针及字符串课本例题  验证上课例题：  **例 void main()**  **{ int a[ ]={5,8,7,6,2,7,3};**  **int y,\*p=&a[1];**  **y=(\*--p)++;**  **printf(“%d ”,y);**  **printf(“%d”,a[0]);**  **}**    二、设计  1、输入两个整型数，交换之后输出（函数，指针）  2、输入3个学生4门课的成绩，查找不及格的学生成绩，并输出。  3、设计并测试一个函数，搜索第1个函数形参指向的字符串，在其中查找第2个函数形参指定的字符首次出现的位置。如果成功，该函数返回指向该字符的指针，如果没有找到指定字符，则返回空指针。  **实验步骤：** 第一步：新建文件夹，保存文件，如图1-1所示   图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一：输入两个整型数，交换之后输出（函数，指针） #include<stdio.h>  void fun(int \*p,int \*p1)//定义形参为两个指针型变量  {  int t;//定义一个中间数  t=\*p;  \*p=\*p1;  \*p1=t;//通过指针将两个数进行交换  }  int main()  {  int a,b;//定义两个整型数  printf("请输入两个整数：");  scanf("%d,%d",&a,&b);//让用户输入这两个数  printf("交换前的结果为：a=%d,b=%d\n",a,b);//输出交换前的结果  fun(&a,&b);//调用交换函数，实参为两个数的地址  printf("交换后的结果为：a=%d,b=%d",a,b);//输出交换后的值  } 实验结果为：  程序二：输入3个学生4门课的成绩，查找不及格的学生成绩，并输出 #include<stdio.h>  int zd(const int \*p,int i)  {  int k,j;  for(k=0;k<4;k++)  if(\*(p+k)<60)//如果学生的成绩小于60，p+k为学生每门成绩的地址  {  printf("第%d个学生的不及格成绩为：",i);  printf("%d\n",\*(p+k));//则输出不及格成绩，  }  }  int main()  {  int a[3][4];//定义一个二维数组，存放三个学生的4门成绩  int i,j;  for(i=0;i<3;i++)  {  printf("请输入第%d个学生的4门成绩：",i+1);  for(j=0;j<4;j++)  scanf("%d",&a[i][j]);  }//输入学生的成绩  for(i=0;i<3;i++)  zd(a[i],i+1);//以每个学生的成绩和第几个学生为实参  } 运行结果为：  程序三：设计并测试一个函数，搜索第1个函数形参指向的字符串，在其中查找第2个函数形参指定的字符首次出现的位置。如果成功，该函数返回指向该字符的指针，如果没有找到指定字符，则返回空指针。 #include<stdio.h>  int fun(char c[],char ch)  {  int i=0;  for(i=0;;i++)  {  if(c[i]==ch)  return &c[i];//返回第一次出现字符的地址  if(c[i]=='\0')  break;//如果没有，则退出循环  }  return 0;//返回0  }  int main()  {  int f;//定义一个存放返回值的变量  char c;//定义一个字符  char ch[100];//定义一个数组  printf("请输入一个字符串：");  gets(ch);//让用户输入一个字符串  printf("请输入一个字符：");  scanf("%c",&c);//让用户输入一个字符  f= fun(ch,c);//  printf("%d\n",f); 运行结果为： | | | | | |
| **实验结果分析及结论**  1：要注意指针自加的话，应为，加上一个自身类型的字节  2：指针地址自加后，地址应进行初始化  3：指针运用时，一定要对齐进行初始化  4：函数传递时要注意时值传递还是地址传递，如果是值传递，则不改变原本的值，反之，则通过地址直接改变其值  5：区分字符串的输入以及字符的输入 | | | | | |
| 1：要注意指针自加的话，应为，加上一个自身类型的字节  2：指针地址自加后，地址应进行初始化  3：指针运用时，一定要对齐进行初始化  4：函数传递时要注意时值传递还是地址传递，如果是值传递，则不改变原本的值，反之，则通过地址直接改变其值  5：区分字符串的输入以及字符的输入 | | | | | |

# 实验报告8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 结构体及文件编程 | | | | |
| 实验地点 | **519** | 实验日期 | 2018.1.8 | 成绩 |  |
| 实验目的 | 通过编程练习，熟练掌握结构体及文件的用法 | | | | |
| 实验原理 | 结构体的定义、初始化及其引用；文件的打开与关闭及读写。 | | | | |
| 实验设备\软件\平台等 | 计算机，win7操作系统，CodeBlock编程环境 | | | | |
| 实 验 内 容 与 实 验 记 录（拓扑图\配置图\流程图\线路图\效果图\代码(段)\运行结果\实验步骤等） | | | | | |
| **实验要求：**  一、验证：结构体及文件课本例题  二、设计  1、编写程序实现以下功能：班委会改选，班长的候选人有三个，统计候选人的选票，并输出班长信息。  2、将一个整数按字节输出。  3、编写程序，统计文本文件中的字符数量、单词数量及行数。要求程序从命令行中获取文件名。 实验步骤： 第一步：新建文件夹，保存文件，如图1-1所示    图1-1 创建文件夹结果  第二步：打开Code::Blocks，新建一个C源文件，如图1-2所示：    图1-2 创建C源程序 程序一：编写程序实现以下功能：班委会改选，班长的候选人有三个，统计候选人的选票，并输出班长信息。 #include<stdio.h>  struct xuanpiao  {  char name[20];  int num;//定义一个结构体，成员为姓名及一个数  };  int main()  {  int i,n;  struct xuanpiao stu[3]={{"张三",0},{"李四",0},{"王五",0}};//定义这个结构体为数组，并对其进行初始化  printf("支持谁就回复对应的序号:\n1:张三 2:李四 3:王五\n");//给予用户提示  for(i=1;i<=5;i++)  {  h:scanf("%d",&n);//让用户进行输入  if(n==1)  stu[0].num++;  else if(n==2)  stu[1].num++;  else if(n==3)  stu[2].num++;//如果满足相应的条件，则相应的计数器加1  else  {  printf("输入错误，请重新输入：");  goto h;//输入其他则返回  }  }  printf("%15s%15s%15s\n","张三","李四","王五");  printf("%15d%15d%15d",stu[0].num++,stu[1].num++,stu[2].num++);//显示出每个人的得票数  int max=stu[0].num;//定义一个最大值，初值为第一个学生的票数  int j=0;  for(i=1;i<3;i++)  {  if(max<stu[i].num)  max=stu[i].num,j=i;//如果后面的同学票数大于此同学，则将后面的这个值赋给max，并将其下标付给j  }  printf("\n最终获选人为：%s",stu[j].name);//输出最终获选人  return 0;  } 实验结果：  程序二：将一个整数按字节输出 #include<stdio.h>  union  {  int a;  char ch[4];  }p;//定义一个共用体，里面有两个成员  int main()  {  int i;  p.a=4;//将共用体中的整型数a赋值为4  for(i=0;i<4;i++)  printf("%d\n",p.ch[i]);//输出共用体中字符型数组的每个值  return 0；  } 运行结果为：  程序三：编写程序，统计文本文件中的字符数量、单词数量及行数。要求程序从命令行中获取文件名 #include<stdio.h>  #include<string.h>  int main()  {  FILE \*fp;  char file\_name[20],ch;//定义个字符数组  int num=1,num1=1,num2=0;//定义三个计数器  printf("输入文件名：");  gets(file\_name);//让用户输入文件名  if((fp=fopen(file\_name,"r"))==NULL)//如果文件不存在  { printf("文件打开失败");//则打开失败  exit(0);  }  while((ch=getc(fp))!=EOF)//循环到文件结尾  {  num++;//统计字符数量  if(ch>='a'&&ch<='z'||ch>='A'&&ch<='Z')  num1++;//统计单词数量  if(ch=='\n')  num2++;//统计行数  }  printf("字符个数为：%d,单词数量为：%d,共有%d行",num-1,num1-1,num2);//显示结果  fclose(fp);//关闭文件  } 文件内容为：  运行结果为: | | | | | |
| **实验结果分析及结论** | | | | | |
| **1：定义联合体后，要知道其成员共用内存，长度为联合体内所占字节最长类型的长度。**  **2：注意打开文件的方式，只读，只写或是其他。**  **3：打开文件后，在最后要将其关闭。**  **4：注意结构体成员的引用方式。** | | | | | |