

【实验目的】

- (1) 掌握 C 语言关系运算符和关系表达式、逻辑运算符和逻辑表达式的使用。
- (2) 熟练掌握各种类型 if 语句的使用方法。
- (3) 掌握 switch 语句以及其中的 break 语句的使用方法。
- (4) 掌握条件运算符和表达式的使用。
- (5) 掌握基本输入输出函数的使用,能正确调用 C 语言提供的数学函数(math.h)和常用字符函数 ctype.h)。
- (6) 掌握简单的单步调试、断点调试和使用 Debug 工具栏调试程序的方法。

【实验内容】

1. 调试样例

使用单步调试、断点调试和 Debug 工具栏调试程序的方法,改正下列程序中的错误。
输入参数 a 、 b 、 c ,求一元二次方程 $ax^2+bx+c=0$ 的根。

有错误的源程序 error3_1.c:

```
#include<stdio.h>
#include<math.h>
int main()
{ double a,b,c,d;
  printf("Enter a,b,c:");
  scanf("%f%f%f",&a,&b,&c);
  d=b*b-4*a*c;          /* 调试时设置断点 */
  if(a=0)
  { if(b==0)
    { if(c==0)
      printf("参数都为零,方程无意义!\n");
    else
      printf("a 和 b 为 0,c 不为 0,方程不成立\n");
    }
  }
```

```

else
    printf("x=%0.2f\n",-c/b);
}
else
    if(d>=0)                /* 调试时设置断点 */
    { printf("x1=%0.2f\n", (-b+sqrt(d))/(2*a));
      printf("x2=%0.2f\n", (-b-sqrt(d))/(2*a));
    }
    else
    { printf("x1=%0.2f+%0.2fi\n", -b/(2*a), sqrt(-d)/(2*a));
      printf("x2=%0.2f-%0.2fi\n", -b/(2*a), sqrt(-d)/(2*a));
    }
return 0;                  /* 调试时设置断点 */
}

```

(1) 打开源程序 error3_1.c,对程序进行编译和连接,没有出现错误和警告信息。但运行程序时,在弹出的运行窗口中输入 a、b、c 的值(2.1 8.9 3.5)后按 Enter 键,发现运行结果(如图 1-3-1 所示)显然错误,说明程序存在逻辑错误,需要调试修改。

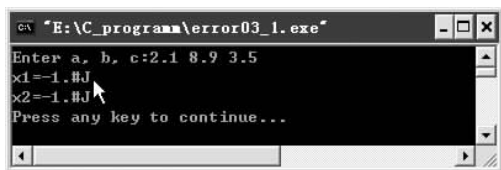


图 1-3-1 程序 error3_1.c 运行窗口

(2) 调试步骤如下。

首先介绍断点的使用。断点的作用就是使程序执行到断点处暂停,用户可以观察当前变量或表达式的值。设置断点,最方便快捷的方法是将鼠标移到代码区中某一条代码的左边(灰色区域),光标由 I 字形变成黑色圆形断点形状(如图 1-3-2 所示),然后单击,看到红色

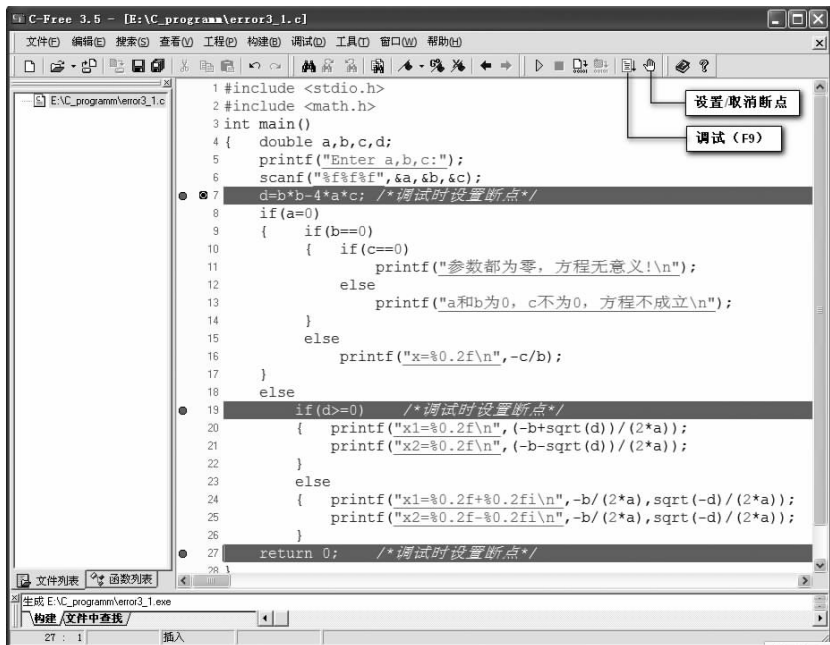


图 1-3-2 在程序 error3_1.c 中设置三个断点

断点就设置完成;另一方法是先将光标点到你想要设置的行,然后单击工具栏上的“设置/取消断点”按钮(如图 1-3-2 所示)。对于已经设置断点的行,对该行重复进行上面的设置断点的操作,将取消断点。

① 调试程序开始,设置三个断点(如图 1-3-2 所示),具体位置见源程序的注释。

② 单击工具栏上的“调试”按钮或按 F9 键,程序开始调试。一旦程序开始调试,C-Free 会自动显示 Debug 工具栏(如图 1-3-3 所示)。

注意: 程序开始调试,执行到某一个断点前,这时“调试”按钮的功能变为“继续”,单击该按钮,程序从该断点处继续执行,直到碰到下一个断点。

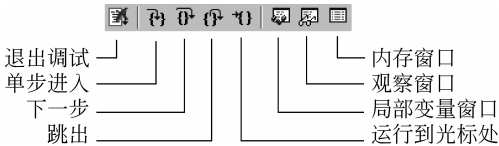


图 1-3-3 Debug 工具栏

③ 在弹出的运行窗口中输入 a、b、c 的值(2.1 8.9 3.5)后按 Enter 键,程序执行到并停在第 1 个断点处,单击 Debug 工具栏上的“局部变量窗口”按钮(如图 1-3-3 所示),然后在局部变量窗口(如图 1-3-4 所示)中查看变量 a、b、c 的值,此时,这些变量的值与输入的值不一致。由此可知第 1 个断点处之前肯定有错误发生,且变量 a、b、c 的取值不正确,应该检查输入函数 scanf 是否正确。仔细检查本程序中的 scanf 函数,发现 scanf 函数中 double 型数据的输入格式控制说明 %lf 错写成 %f 了。

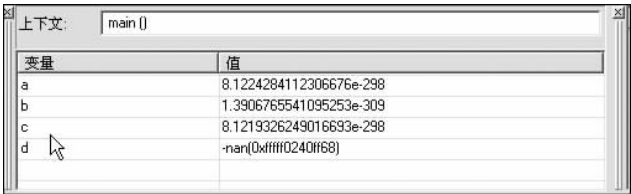


图 1-3-4 局部变量窗口

注意: 当程序处于调试停止状态,局部变量窗口将显示当前运行环境下,所有局部变量的值。对于源程序 error3_1.c 的程序运行状态,有三个局部变量。局部变量窗口如图 1-3-4 所示,其中上下文显示的是当前程序运行的函数环境,包括参数的值。

④ 单击 Debug 工具栏上的“退出调试”按钮(如图 1-3-3 所示)或按 Ctrl+F9 组合键,结束程序调试。然后将本程序中的输入语句改为“scanf(“%lf%lf%lf”,&a,&b,&c);”。改正后,重新对本程序进行编译和连接。再单击工具栏上的“调试”按钮,在弹出的运行窗口中同样输入 a、b、c 的值(2.1 8.9 3.5)后按 Enter 键。程序执行到第 1 个断点处,同样在局部变量窗口中查看变量 a、b、c 的值。此时,这些变量的值与输入的值一致(如图 1-3-5 所示)。但在图 1-3-5 所示的局部变量窗口中,查看到此时变量 d 的值显然不正确,原因是程序执行到第 1 个断点处时,断点处的语句并未执行。

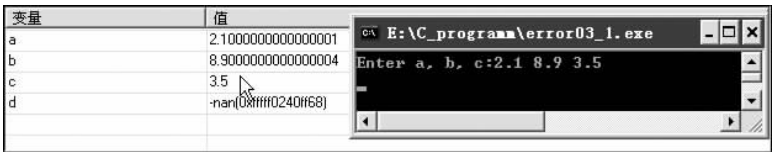


图 1-3-5 局部变量窗口中变量的值与输入的值对照

⑤ 单击 Debug 工具栏上的“下一步”按钮(如图 1-3-3 所示)或按 F8 键后,在图 1-3-6 所示的局部变量窗口中观察到变量 d 的值变为 49.81,此时,d 值是正确的。“下一步”按钮的功能是单步执行,即单击一次执行一行(如图 1-3-6 所示),编辑窗口中的箭头指向某一行,表示程序将要执行该行。单击 Debug 工具栏上的“观察窗口”按钮(如图 1-3-3 所示),即可打开“观察”窗口(如图 1-3-6 所示)。右击“观察”窗口,弹出菜单,选择“添加观察”,弹出“添加观察”对话框(如图 1-3-7 所示),在编辑框输入需要观察的表达式,就可以实时地观察这个表达式的值。

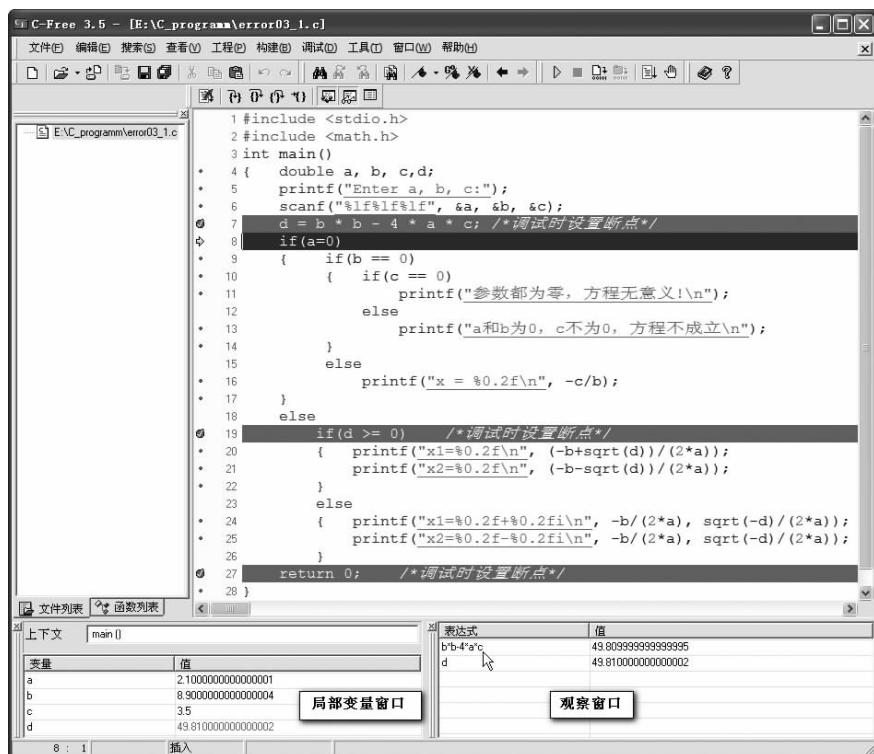


图 1-3-6 单步调试程序 error3_1.c 时所对应的各窗口状态

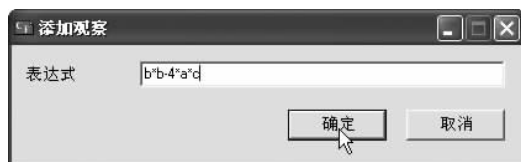


图 1-3-7 “添加观察”对话框

⑥ 继续单击工具栏上的“调试”按钮,程序运行到第二个断点(如图 1-3-8 所示),在局部变量窗口中观察到变量 d 的值是 49.81,说明方程有实根。但发现变量 a 的值发生变化,值变为 0,显然程序有错。仔细分析源程序,发现图 1-3-8 中的第 8 行把 if(a==0)写成 if(a=0)。二者的区别是: if(a==0)表示判断 a 是否等于 0,当 a 的值为 0 时,a==0 表达式的值为 1(表示“真”),而 a=0 表达式的值为 0(表示“假”),即 if(a=0)表示把 0 值赋给 a,然后判

断 a 值为“假”，从而执行 if(a=0) 语句的 else 分支语句。因此编写程序时一定要分清是用关系运算符==还是用赋值运算符=。

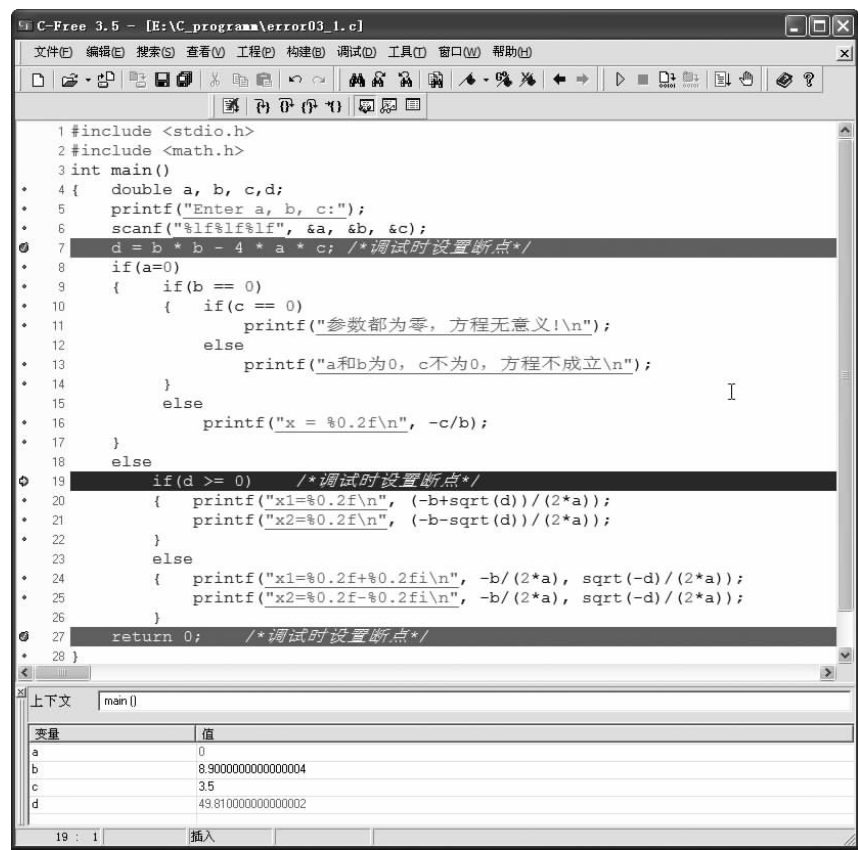


图 1-3-8 运行到第 2 个断点处所对应的编辑窗口和局部变量窗口

⑦ 单击 Debug 工具栏上的“退出调试”按钮，结束程序调试，回到编辑状态。修改上述错误后，重新编译和连接。然后单击工具栏上的“调试”按钮，在弹出的运行窗口中同样输入 a、b、c 的值(2.1 8.9 3.5)后按 Enter 键，程序执行到第 1 个断点处，再次单击工具栏上的“调试”按钮，此时变量 a、b、c、d 值均为正确。然后再单击“调试”按钮，程序运行到最后一个断点，运行窗口显示 $x_1 = -0.44$, $x_2 = -3.80$ ，符合题目要求。

⑧ 单击 Debug 工具栏上的“退出调试”按钮，结束程序调试，回到编辑状态。将鼠标分别移到代码区中 3 个断点所在行的左边(灰色区域)，光标由 I 字形变成黑色圆形断点形状，然后单击，即可取消断点。也可以先将光标分别点到三个断点所在行，然后单击工具栏上的“设置/取消断点”按钮(如图 1-3-2 所示)，也可取消断点。取消已设置的断点后，再次运行程序，在弹出的运行窗口中输入不同的 a、b、c 值(2.1 3.2 3.5)后按 Enter 键，运行窗口显示 $x_1 = -0.76 + 1.04i$, $x_2 = -0.76 - 1.04i$ ，符合题目要求。

注意：本程序中 printf 函数中的“%0.2f”等价于“%.2f”，表示输出结果保留 2 位小数，因输出项的指定宽度为 0，肯定小于数据实际宽度，所以数据整数部分不受指定宽度的限制，应完整输出。另外，为了检查调试后的本程序是否正确，根据题意，输入参数至少应该有

5 组:

- ① a、b、c 都为 0;
- ② a 和 b 为 0,c 不为 0;
- ③ a 为 0,b 不为 0,c 任意;
- ④ a 不为 0,且 a、b、c 满足 $b^2-4ac\geq 0$;
- ⑤ a 不为 0,且 a、b、c 满足 $b^2-4ac<0$ 。

请输入不同的参数,观察运行结果是否都符合题目要求。

2. 程序修改题

模仿以上调试样例,使用单步调试和断点调试程序的方法改正下列程序中的错误。

(1) 输入实数 x , 计算并输出下列分段函数 $f(x)$ 的值, 输出时保留 2 位小数。

$$y = f(x) = \begin{cases} \frac{1}{x} & x = 10 \\ x & x \neq 10 \end{cases}$$

有错误的源程序 error3_2.c:

```
#include<stdio.h>
int main()
{ double x,y;
  printf("Enter x:\n");
  scanf("%f",x);
  if(x=10)
  { y=1/x
  }
  else(x!=10)
  { y=x;
  }
  printf("f(%.2f)=%.1f"x y);
  return 0;
}
```

提示:

- ① 语句是否都有分号结束。
- ② 赋值运算符=与关系运算符==的区别。
- ③ 使用输入输出函数时,各参数的书写形式要正确,且 scanf 函数的格式控制字符串中尽量不要出现普通字符,若有,在输入数据时也需要原样输入。
- ④ double 型数据的输入格式控制说明必须使用 %lf,而输出格式控制说明可以用 %f。

(2) 输入三角形的三条边的边长 $a、b、c$,如果能构成一个三角形,输出面积 area 和周长 perimeter(保留 2 位小数);否则,输出 These sides do not correspond to a valid triangle。

在一个三角形中,任意两边之和大于第三边。三角形面积计算公式如下,其中 $s=(a+b+c)/2$ 。

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

有错误的源程序 error3_3.c:

```
#include<stdio.h>
#include<math.h>
int main()
{
    double a,b,c;
    double area,perimeter,s;
    printf("Enter 3 sides of the triangle:");
    scanf("%lf%lf%lf",&a,&b,&c);
    if(a+b>c||b+c>a||a+c>b);
        s=(a+b+c)/2;
        area=sqrt(s*(s-a)*(s-b)*(s-c));
        perimeter=a+b+c;
        printf("area=%.2f;perimeter=%.2f\n",area,perimeter);
    else
        printf("These sides do not correspond to a valid triangle\n");
    return 0;
}
```

提示:

① 算术运算符、赋值运算符、关系运算符、逻辑运算符参加运算的先后顺序是: 逻辑非运算符→算术运算符→关系运算符→逻辑与运算符→逻辑或运算符→赋值运算符。关系表达式和逻辑表达式的使用。

② 注意逻辑与(&&)和逻辑或(||)运算符的含义及逻辑运算的规则。

③ if...else 语句中 if 表达式后面不能有分号(;)。

④ if 语句中若包括多条语句,一定要用大括号({})括起来构成复合语句。

扩展: 根据输入的三个边长 a 、 b 、 c , 判断它们能否构成三角形;若能构成三角形,继续判断该三角形是等边、等腰还是一般三角形。应如何编程实现?

(3) 求输入一个字符,判断它是空格字符、数字字符、小写字母、大写字母,还是其他字符。

有错误的源程序 error3_4.c:

```
#include<stdio.h>
int main()
{
    char c;
    printf("Enter a character:");
    c=getchar();
    if(c==' ')
        printf("This is a space character\n");
    else if(0<=c<=9)
        printf("This is a digit\n");
    else if(c>=a&&c<=z)
        printf("This is a small letter\n");
    else if(c>='A'&&c<='Z')
        printf("This is a capital letter\n");
}
```

```

else
    printf("This is an other character\n");
return 0;
}

```

提示:

① 注意空字符('')和空格字符(' ')、数字字符('0')和数字(0)、字符常量('a')和变量(a)的区别。

② C 语言中关系表达式和逻辑表达式的值只有两种结果,要么为“真”(用 1 表示),要么为“假”(用 0 表示)。因而要想判断一个字符是否为数字字符,则 if 语句中的表达式必须用 $c >= '0' \& \& c <= '9'$ 表示。

③ 注意 if...else if 语句、if 语句中的嵌套关系和匹配原则。

扩展: 使用 ctype.h 中的函数重做此题。

(4) 下面是进行加、减、乘、除、求余运算的程序,用户输入一个形式如“运算数 运算符 运算数”的表达式,计算两个整数相应的值,最后输出结果。

有错误的源程序 error3_5.c:

```

#include<stdio.h>
int main()
{
    char c;
    int a,b;
    printf("输入 a 运算符 b:");
    scanf("%d%c%d",&a,&c,&b);
    switch(c);
    {
        case '+': printf("%d+%d=%d\n",a,b,a+b);break;
        case '-': printf("%d-%d=%d\n",a,b,a-b);
        case '*': printf("%d*%d=%d\n", a,b,a*b);break;
        case '/': if(b!=0) printf("%d/%d=%d\n",a,b,a/b);
                    else printf("error!");
                    break;
        case '%': if(b!=0) printf("%d%%d=%d\n",a,b,a%b);
                    else printf("error!");
                    break;
        default: printf("运算符输入错误!\n");
    }
    return 0;
}

```

提示:

① 要利用 printf 函数输出显示 % 符号必须要有两个 %% 符号,其中第一个 % 符号是转义字符。注意: % 符号还可作为求余运算符。

② 注意 switch 语句格式及使用方法,包括 switch 语句中表达式后面不能有分号、switch 语句中 break 语句的作用等。

③ 在运行窗口输入数据时应注意: 两个整数与运算符之间都应该无任何符号。

扩展：如何用 if...else if 语句实现以上程序功能？要求编写出的程序同样是有较好的健壮性，例如，处理输入的运算符为非法情况、除数为 0 等情况。

3. 程序填空题

完善下列程序，然后调试运行程序。

(1) 输入一个小写字母，将字母循环后移 5 个位置后输出，例如 ‘a’ 变成 ‘f’，‘w’ 变成 ‘b’。

有待完善的源程序 fill3_1.c:

```
#include<stdio.h>
int main()
{   char c;
    c=getchar();
    if( ① ) c=c+5;
    else if (c>='v' && c<='z') ② ;
    putchar(c);
    return 0;
}
```

提示：注意 if 语句中的嵌套关系和匹配原则，字母循环后移多个位置的逻辑关系。

(2) 输入 2 个数，输出其中较大的数。

有待完善的源程序 fill3_2.c:

```
#include<stdio.h>
int main()
{   int a,b,max;
    printf("Enter two integers:");
    ① ;
    max= ② ;
    printf("\nmax= ③ \n", max);
    return 0;
}
```

提示：注意输入输出函数、条件运算符和表达式的使用。

条件运算符由 ? 和 : 组成，它需要三个运算量。结合方向是自右至左。

条件表达式的一般形式是：“表达式 1? 表达式 2: 表达式 3;”，其含义是当表达式 1 的值为非 0 时，以表达式 2 的值作为条件表达式的值，否则，以表达式 3 的值作为条件表达式的值。由于赋值运算符的优先级比条件运算符低，语句“max=a>b?a:b;”相当于“max=(a>b?a:b);”，因此等价于：“if(a>b)max=a; else max=b;”。

4. 程序设计题

(1) 输入任意三个实数 n1、n2、n3，求其中最大的一个数，输出时保留 2 位小数。

扩展：

① 如果要在三个字符中找出最小的一个，如何修改程序？

② 如果要把这三个数按从小到大的顺序输出,应如何编程实现?

(2) 输入一个整数,如果是偶数,则输出 Even number,如果是奇数,则输出 Odd number。

扩展:

① 输入两个整数,判断第一个数是否是第二个数的倍数,如果是,则输出 Yes,否则输出 No。

② (鸡兔同笼)一个笼子里面关了鸡和兔子(鸡有 2 只脚,兔子有 4 只脚,无例外)。已经知道了笼子里面脚的总数 a ,问笼子里面至少有多少只动物,至多有多少只动物。

提示: 鸡兔同笼问题是笼子里至少有多少只动物,至多有多少只动物。显然在动物脚总数一定的情况下,要使笼子里的动物数量最多,就要使每个动物的脚数尽可能少,反之一样,所以当笼子中动物数达到最多时,笼子里的动物应都是鸡,而当笼子中动物数达到最少时,应都是兔子或多只兔子加一只鸡,并且动物脚的个数一定是偶数。也就是说,对于给定的整数 a ,如果 a 是奇数,表示没有满足要求的情况出现,则输出: 0 0(第一个数表示最少的动物数,第二个数表示最多的动物数,下同);如果 a 是 4 的倍数,则输出: $a/4$ $a/2$;如果 a 不是 4 的倍数,则输出: $a/4+1$ $a/2$ 。

(3) 输入一个整数 n ,判断 n 能否被 3、5、7 整除,并输出以下信息之一。

① 能同时被 3、5、7 整除。

② 能被其中两数整除。

③ 能被其中一个数整除。

④ 不能被 3、5、7 任一个数整除。

输入输出样例如下(运行 4 次)。

第一次运行:

Enter n:15

能被其中两数整除

第二次运行:

Enter n:14

能被其中一个数整除

第三次运行:

Enter n:105

能同时被 3、5、7 整除

第四次运行:

Enter n:17

不能被 3、5、7 任一个数整除

(4) 如果输入一个 1 至 7 之间的数字,则输出星期一至星期日的英文单词,否则输出 error。

(5) 输入五级制成绩(A~E),输出相应的百分制成绩(0~100)区间。五级制成绩对应的百分制成绩区间为: A(90~100)、B(80~89)、C(70~79)、D(60~69)、E(0~59)。

要求: 分别用 if 语句和 switch 语句实现。

提示: 该程序应该运行 6 次,每次测试一种情况,即分别输入 A、B、C、D、E 和其他字符。

扩展：若要求根据输入的学生成绩，将测试的分数自动转变成对应的等级。例如，输入 93，则输出 A。应如何编程实现？同样要求分别用 if 语句和 switch 语句实现。

(6) 输入实数 x ，求下列分段函数 $f(x)$ 的值，结果保留 3 位小数。

$$f(x) = \begin{cases} \frac{8}{x^2 + x + 1} & -5 \leq x < 0 \\ \frac{7}{x^2 + x + 1} & 0 \leq x < 5 \\ \frac{2}{x + 8} & 5 \leq x < 10 \\ 0 & x \geq 10, x < -5 \end{cases}$$

(7) 输入一个字符，如果是数字字符，则转换成对应的数字后输出，否则输出不是数字字符的提示说明。

输入输出样例如下(运行 2 次)。

第一次运行：

Enter a character: 5

Character: '5' Digit:5

第二次运行：

Enter a character: a

'a' is not the numeric character!

扩展：

① 要求用条件运算符和表达式实现以上程序功能。应如何编程实现？

② 输入一个字符。如果是大写字母，则将其转换成对应的小写字母后输出；如果是小写字母，则将其转换成对应的大写字母后输出；否则输出不是字母的提示说明。要求使用字符函数判断和转换。

提示：

① 转义字符的输出方法。例如，要想输出显示单引号'，必须在 printf 函数中使用\'。

② 使用字符函数判断和转换时，在程序的开头加命令行 #include <ctype.h> 引入 ctype.h 头文件。

③ 能正确调用 C 语言提供的常用字符函数：isupper(c) 判断字符变量 c 是否为大写字母，等价于 if(c >= 'A' && c <= 'Z')；islower(c) 判断 c 是否为小写字母，等价于 if(c >= 'a' && c <= 'z')；isdigit(c) 判断 c 是否为数字，等价于 if(c >= '0' && c <= '9')；tolower(c) 表示将 c 中的字母转换成小写字母；toupper(c) 表示将 c 中的字母转换成大写字母。

(8) 输入月薪 salary，输出应交的个人所得税 tax(保留 2 位小数)。按照 2011 年开始实行的新的个人所得税法，计算公式为：

$$\text{tax} = \text{rate} \times (\text{salary} - 3500) - \text{deduction}$$

当 salary ≤ 3500 时，rate = 0、deduction = 0

当 3500 < salary ≤ 5000 时，rate = 3%、deduction = 0

当 5000 < salary ≤ 8000 时，rate = 10%、deduction = 105

当 8000 < salary ≤ 12 500 时，rate = 20%、deduction = 555

当 12 500 < salary ≤ 38 500 时，rate = 25%、deduction = 1005

当 $38\ 500 < \text{salary} \leq 58\ 500$ 时, $\text{rate} = 30\%$ 、 $\text{deduction} = 2755$

当 $58\ 500 < \text{salary} \leq 83\ 500$ 时, $\text{rate} = 35\%$ 、 $\text{deduction} = 5505$

当 $83\ 500 < \text{salary}$ 时, $\text{rate} = 45\%$ 、 $\text{deduction} = 13505$

提示：该程序应该运行 8 次，每次测试一种情况。

(9) 根据输入的生日(年 y、月 m、日 d)和今天的日期(年 yt、月 mt、日 dt)计算并输出实足年龄。应如何编程实现？

提示：在以下两种情况下，都需要将计算出的年龄 $\text{age}(\text{age} = \text{yt} - y)$ 减 1，才为实足年龄。一是本年还未到出生月，即当前月份(mt)小于生日月份(m)；或者本年已进入生日月份，即当前月份(mt)等于生日月份(m)，但还未到出生日，即当前日(dt)小于出生日(d)。

(10) 输入一个年月，计算并输出该年月有多少天。应如何编程实现？

提示：

① 判断闰年的条件是：能被 4 整除但不能被 100 整除，或者能被 400 整除。

② 闰年 2 月份为 29 天，非闰年 2 月份为 28 天。4 月份、6 月份、9 月份、11 月份为 30 天，其他月份为 31 天。

【实验结果和分析】

(1) 将 C 语言源程序、运行结果写在实验报告上。

(2) 分析源程序和运行结果，并将遇到的问题和解决问题的方法写在实验报告上。