

第一次实验报告

2018.9.22 PB16000702 韦璐

实验目的：1.C 程序的基本结构

2.函数形式与 `main()` 函数，库函数与用户自定义函数

3.库函数 `printf()` 的使用格式和输出结果

4.库函数 `scanf()` 的使用格式和输入方法

5.库函数 `getchar()` 和字符的读取

6.C 程序的编辑，编译和连接

第一题

实验代码：

（要求在实验课之前提前完成，上机只需要调试代码就可以了，因为上机时间很紧张。能力有限的同学也可以不提前完成全部代码，但提交实验报告时要求有完整的代码。）

（范例不用写代码。）

```
#include<stdio.h>
int main()
{
    int a,b,c,max;
    printf("Enter a,b,c: \n");
    scanf("%d%d%d",&a,&b,&c);
    max=a;
    if(max<b)
        max=b;
    if(max<c)
        max=c;
    printf("max=%d\n",max);
}
```

调试分析：

（属于实验时记录部分，也可在完成实验后再整理。主要记录实验中遇到的问题，及分析调试后的结果，是否成功解决问题，否的话最后是如何解决问题的）

（也可以直接在前面的代码部分标注出问题的位置，以及解决方法，如下）

实验中，遇到了两个问题，第一个是我在编译以后发现自己的命名和文件夹都放错了，所以就改了名字换了另外一个文件夹，结果发现文件反复编译就是无法运行，后来发现是路径错了，调回来以后才运行。第二个问题是我在运行的时候输入数字结果要么没反应要么数字变成一排出来了要么就全是错的，后来直接打成空格才好一些。这是因为不熟悉 `scanf` 的输入格式导致的。

scanf 的输入格式

这个函数的功能是接受用户从键盘输入的数据，并按照格式控制符的要求进行类型转换，然后送到地址表列中对应的变量存储单元去。返回值是输入数据的个数。

一般形式为：`scanf(“输入格式串”，输入项地址表列);`

1. 输入格式串的组成一般只使用格式符。

2. C 语言规定输入项必须由地址构成，项与项之间使用逗号分隔。
3. 输入格式串中格式符的个数决定了输入项地址表列中可输入数据的个数，且数据类型必须从左到右一一匹配对应。
4. “&”为地址运算符或称取地址符。变量的地址表示法是在变量前加“&”符号，运算结果为 a 变量在内存中的对应存储单元的地址。
5. 如果输入格式串中加入了格式符以外的其他字符，则在通过键盘等输入设备输入数据的时候，这些也要加入；若没有，就空格或者回车或者制表符作为间隔标志；但是使用%c的时候，不用间隔标志。
6. 用户可以指定输入数据的宽度，但不可以指定小数点的位数，系统自动按照此宽度截取读入的数据。
7. 输入数据遇到空格，回车或者制表符以及其他各种非法输入时，认为该项数据输入结束。
8. 当遇到格式符“%*”时，表示跳过对应的输入数据项不予读入。

实验结果：

（实验前应准备好多组典型数据，用于测试程序）

	Input（实验前准备的典型数据）	Output（测试结果）
1	1 2 3	3
2	2 3 4	4
3	3 4 5	5
.....		

实验总结：

要仔细阅读每一种函数的使用方法，不然会对以后的使用造成困难。

第二题

实验代码：

```
#include<math.h>
#include<stdio.h>
main()
{
    double a,b,c,p,d,x1,x2;
    a=1.0;b=-(1.0e+12+1.0);c=1.0e+12;
    p=sqrt(b*b-4.0*a*c);
    d=2.0*a;
    x1=(-b+p)/d;
    x2=(-b-p)/d;
    printf("x1=%e\nx2=%e\n",x1,x2);
}
```

调试分析：

无

实验结果：已经给定

实验总结：

到了这个实验实在受不了自己之前把错误显示栏取消的操作了，然后就研究了一下这个软件，然后设置一个快捷键现在感觉使用起来方便多了。

第三题

实验代码：

```
#include<stdio.h>
main()
{
    printf("The UNIX System.\n");
    prt( );
}
#include<stdio.h>
prt( ){
    printf("The C Programming Language.\n");
}
```

这里我是直接建在一个项目里面的。

调试分析：一开始主要是编译无法通过，后来知道是用项目以后就好多了。

实验结果：

The UNIX System.

The C Programming Language.

实验总结：

无

习题

第一题

实验代码：

```
#include<stdio.h>
main()
{
    printf(" * * * * * \n");
    printf("          Very Good!          \n");
    printf(" * * * * * \n");
}
```

//这里运行的时候就会发现是对齐的但是复制到 word 就是这样的了

调试分析：

库函数 printf 的使用：这个函数的作用是 anzhao 规定的格式把输出项输出到显示屏上。

一般的形式为：printf(“输出格式串”，输出参数项表列)；

1. 在双引号定界的输出格式串中，可以使用三种不同的字符成分：格式符，转义字符，一般字符。

- (1) 格式符规定了相应输出项的输出形式。常用格式及其意义如下：

`%d` 按十进制整数形式输出

`%x` 按十六进制整数形式输出

`%f` 按浮点数（实数）形式输出

`%c` 按单一字符形式输出

`%s` 按字符串形式输出

.....

格式符的基本形式：

`%[<数据输出宽度说明>]<格式字符>`，也即是说定义输出形式的时候还可以规定其输出项的宽度。例如`%6d`，`%10.2f` 等。“`%m.nf`”表示保留 n 位小数（四舍五入）并且共占用 m 列（含小数点）。数据宽度说明中如果实际数据位小于宽度（绝对值），则宽度大于零时左补空格，宽度小于零时右补空格。如果实际位数大于宽度，那么四舍五入。没有说明就按照实际宽度输出（实数位数按格式符默认位数输出）。

- (2) 转义字符用于输出控制代码和特殊字符，其形式为“`\`”打头后跟一个字母或数字。“`\n`”为回车换行控制代码。或者想要输出任何字符都需要这么做。

- (3) 输出格式串中，格式符和转义字符以外的其他字符是一般字符，可以原样输出。

2. 输出参数可以是常数变量和表达式等，可以有多个输出项，之间用逗号分隔。
`printf` 函数执行的时候会从左到右，依次计算输出参数项表列中各项的值，然后将这些值按照格式输出到显示器上。所以两边的个数必须相同，当格式说明的个数小于输出表列的个数的时候，，多余的输出表列的项不输出。这个函数的返回值是输出数据的个数。

3. 这个函数也可以没有输出项，此时可以利用它输出提示信息。

实验结果：

无

实验总结：

无

第二题

实验代码：

```
#include<stdio.h>
#define PI 3.1415926535
main()
{
    double R,H,V;
    printf("请输入圆柱体的参数：(先输入底面半径再输入圆柱体高度)");
    scanf("%lf%lf",&R,&H);
    V=H*PI*R*R;
    printf("圆柱体的体积为： %lf",V);
}
```

调试分析：

一次通过，除了中文符号以外并没有任何问题。

实验结果：

（实验前应准备好多组典型数据，用于测试程序）

	Input（实验前准备的典型数据）	Output（测试结果）
1	2 3	37.699112
2	3 4	113.097336
3	4 5	251.327412
.....		

实验总结：

无

第三题

实验代码：

```
#include<stdio.h>
#include<math.h>
main()
{
    double c,f;
    printf("请输入华氏温度：");
    scanf("%lf",&f);
    c=(f-32)*5/9;
    printf("摄氏温度为： %lf",c);
}
```

调试分析：

调试正常一次通过，但是发现出现的结果都是错误的，检查发现我定义的变量都是 `double` 类型的但是不论是 `scanf` 还是 `printf` 我都使用 `f` 的格式进行输入输出导致精度不够所以结果出错，`double` 类型的数据应该用 `lf` 进行输出。

实验结果：

（实验前应准备好多组典型数据，用于测试程序）

	Input（华氏温度）	Output（摄氏温度）
1	86	30
2	1.8	1
3	98.6	37
.....		

实验总结：
无

第四题

实验代码：

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
main()
{
    double a,b,c,p,S;
    printf("请输入三条边的边长： ");
    scanf("%lf%lf%lf",&a,&b,&c);
    if(a+b>c&&a+c>b&&b+c>a){
        printf("能构成三角形");
        p=(a+b+c)/2.0;
        S=sqrt(p*(p-a)*(p-b)*(p-c));
        printf("三角形的面积是： %lf",S);
    }
    else{
        exit(0);
    }
}
```

调试分析：

Scanf 输入的时候只用在数据之间有空格就可以了，数据两边不需要
差点忘记计算公式以后要用 printf 来进行输出

注意字母的大小写

公式中有 double 型的数据的时候要注意常数也要有小数点

If else 的使用方法：就是我上面用的方法，但是要注意不要在 if 和 else 之间加上分号，不然就会导致形成一个完整的语句然后就会使得结果没有，也可以没有 else 而只是使用 if 进行判断然后执行程序。

实验结果：

（实验前应准备好多组典型数据，用于测试程序）

	Input（实验前准备的典型数据）	Output（测试结果）
1	6 7 8	20.333163
2	8 9 10	43.197039
3	1 5 6	无
.....		

实验总结：无

第五题

实验代码：

```
#include<stdio.h>
main()
{
    char c1,c2,c3,c4;
    scanf("%c%c",&c1,&c2);
    getchar();
    c3=getchar();c4=getchar();
    printf("%c %c %c %c\n",c1,c2,c3,c4);
    printf("%c,%d\n",c1,c1);
    printf("%d,%d\n",sizeof(c1),sizeof(1234));
}
```

调试分析：

不容易发现的没有编译报错的错误：

输出格式串中的符号，数据的类型，输出格式串中输出数据的类型，**double** 等类型在数字的地方没有加小数点。

实验结果：

输入

12

3456

输出

1 2 3 4

1, 49

1, 4

实验结果分析：

可以看到这个程序中使用的函数 **getchar** 是为了让我们输入字符，也就是我们的 ASCII 码上左边的部分，但是在两行输入之间还有一个 **getchar**，我现在猜测是为了接受回车键的，然后我们可以看到对第一个输出 **c3=3,c3=4**，就可以发现其实 **getchar** 并没有接受我们输入的超过长度的字符。第二个输出我们可以发现当我们按照字符输出的时候它会输出字符，但按照整型输出的时候会输出这个字符代表的数字，也就是说电脑默认 **1** 是一个字符，最后一个输出说明了字符的长度是一个数字。

实验总结：

1.执行 **getchar** 函数的时候，首先直接从缓冲区读取字符，直到输入缓冲区为空才等待从键盘继续输入。一行上输入的所有字符一并输入缓冲区，输入完毕，函数从缓冲区读取字符，只要缓冲区还有剩余的字符，后面的函数就会直接从中读取。回车和空格都属于字符。

2.**sizeof** 是长度运算符，对象可以是常量变量表达式和数组名等，运算结果是运

算对象存储字节长度，对表达式而言是表达式值的字节长度。

第六题

实验代码

```
#include<math.h>
#include<stdio.h>
main()
{
    float a,b,c,p,d,x1,x2;
    a=1.0;b=-(1.0e+12+1.0);c=1.0e+12;
    p=sqrt(b*b-4.0*a*c);
    d=2.0*a;
    x1=(-b+p)/d;
    x2=(-b-p)/d;
    printf("x1=%e\nx2=%e\n",x1,x2);
}
```

实验结果

只有 e2 变成了完全是零

实验结果分析：

我觉得很有可能是因为我这里要让实数参与四则运算，计算机会先将数字转变成二进制然后再运算最后才转变成十进制数发给我，我试了一下如果不参与运算，浮点数是可以准确输出的，所以我觉得浮点数在考虑精确度的情况下是非常适合进行数值计算的，但是无法精确计算。