※scanf()模式匹配的原理

参考：<https://www.cnblogs.com/shuaishuaidefeizhu/p/5886899.html>

一、理论说明

1、用户**输入**的字符，会以ASCII码形式存储在键盘缓冲区；//如果字符接收到换行符，输出%c || %d 得到10（换行符的ASCII码） 见程序实例2

2、每调用一次**scanf函数**

①就从键盘缓冲区读走一个字符，相当于清除缓冲区；

①’调用scanf()函数时，用户最后输入的回车也会储存在键盘缓冲区；//见程序示例2

②若用户一次输入n个字符，则前n次调用scanf函数都不需要用户再次输入，直到把缓冲区的数据全部读取(清除)干净

二、代码示例

三、清除缓冲区的几种方法

1、为什么要清理？

我们使用多个scanf()的时候,如果输入缓冲区还有数据的话,那么scanf()就不会询问用户输入,而是直接就将输入缓冲区的内容拿出来用了,这就导致了前面的错误影响到后面的内容(比如scanf()输入完后会有换行符残留，会影响下一个scanf(()、getchar()、gets()的输入!————标志：多次(2+)输入、多次取缓冲区

为了隔离这种问题,需要通过各种方法将输入缓冲区的内容读出来（清除）

1、fflush(stdin)

//程序示例1错误！在试验区进行了验证与更改 见 scanf()原理程序实例2.c

1. 此种方法对vc可以，但对xcode和linux不适用

② 缓冲区的内容会全部清空！

2、while+getchar

while (ch=getchar() != '\n' && ch != 'EOF'),直到读取到缓冲区的换行或者空值

由于getchar()取到了有效的一个字符，所以要用ch保存，（while的出口是其条件的否命题）

3、利用模式匹配

scanf(“ %d”, &a);

在要输入的东西前多加一个空格以匹配空符！

//详细代码见 试验区有关缓冲区的实验.c

int a, c; char b;

scanf("%d %c", &a, &b); //若直接输入一个数字 和一个长度超过一的 数字 或 一个字符加一个数字

printf("1\n"); //会直接跳过第二个scanf()的步骤

scanf("%d",&c); //因为缓冲区里已存在符合模式的内容

printf("%d %c %d", a, b, c); // 这就是模式匹配的原理

缓冲区在实践区的实验.c

标志：多次(2+)输入

利用模式匹配的方法解决缓冲区的问题

float x1, x2, y1, y2;

float d;

x1=x2=y1=y2=d=0.00;

printf("Please input the coordinators of two points: \n");

printf("point1(x1, y1): ");

scanf("(%f, %f)",&x1, &y1);

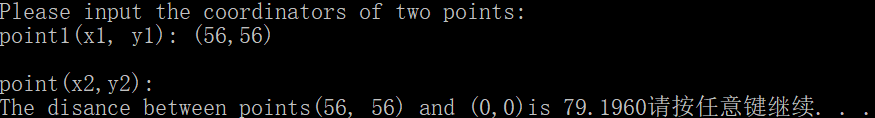
printf("\npoint(x2,y2):");

scanf(" (%f, %f)", &x2, &y2);

d= sqrtf((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

printf("\nThe disance between points(%g, %g) and (%g,%g)is %.4f", x1, y1, x2, y2, d);

//如果第二个scanf的格式串里没有前面的若干（至少一个）个空格，会出现下面的状况



p.s. 如果在终端输入的时候用中文输入法也会出现此问题