

第3章 键盘输入与屏幕输出

——数据的格式化键盘输入

本节要讨论的主要问题

- 如何按指定格式从键盘输入指定类型的数据？



数据的格式化键盘输入

格式

`scanf` (格式控制字符串, 输入地址表);

`scanf ("%d%f", &a, &b);`

转换说明

可变长度输入
地址参数表

&是取地址运算符



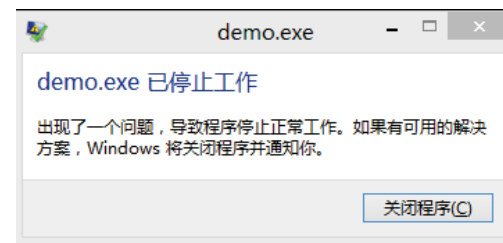
输入数据的格式控制

```
#include <stdio.h>

main()
{
    int a, b;

    scanf("%d %d", a, b);
    printf("a = %d, b = %d\n", a, b);
}
```

12 34 ✓
a = 12, b = 34



问题：若程序第5行语句未使用`&`，结果会怎样？

warning: 'a' is used uninitialized in the function

warning: 'b' is used uninitialized in the function



输入数据的格式控制

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int  a, b;
```

```
    scanf("%f %f", &a, &b);
```

```
    printf("a = %d, b = %d\n", a, b);
```

```
}
```

问题：若程序第5行语句**格式字符写错**，结果会怎样？

结果：scanf() 将读入错误的数据

12 34✓

a = 1094713344, b = 1107820544



输入数据的格式控制

■ 格式

`scanf` (格式控制字符串, 输入地址表);

`scanf ("%d%f", &a, &b);`

如何分隔多个输入数据?

1) 空格、Tab或回车

2) 达到输入位宽

3) 遇非法字符

遇以上情况都认为一个数据输入结束:

输入数据的格式控制

```
#include <stdio.h>
main()
{
    int a, b;

    scanf("%d %d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

```
12✓
34✓
a = 12, b = 34
```

```
12 34✓
a = 12, b = 34
```

问题：当要求程序输出结果为

a = 12, b = 34

时，用户应该如何输入数据？



输入数据的格式控制

```
#include <stdio.h>

main()
{
    int a, b;

    scanf("%2d%2d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

```
1234 ✓
a = 12, b = 34
```

问题：当要求程序输出结果为

$a = 12, b = 34$

时，用户应该如何输入数据？



输入数据的格式控制

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int  a, b;
```

```
    scanf ("%d %d", &a, &b);
```

```
    printf ("a = %d, b = %d\n", a, b);
```

```
}
```

12 3a ✓

a = 12, b = 3

问题：如果用户输入了非数值的字符，例如，输入了12 3a，那么程序运行结果如何？



输入数据的格式控制

```
#include <stdio.h>

main()
{
    int a, b;

    scanf("%d %d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

问题：如果用户输入的是123a，那么结果又会如何呢？

```
123a ✓
a = 123, b = 121
```



输入数据的格式控制

```
#include <stdio.h>
main()
{
    int a, b;

    scanf("%d %d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

如何判断函数scanf()
是否成功读入了指定的
数据项数呢？

问题：如果用户输入的是123a，那么结果又会如何呢？

```
123a✓
a = 123, b = 121
```



输入数据的格式控制

```
#include <stdio.h>
main()
```

```
{
```

```
    int  a, b;
```

scanf () 的返回值=正确读入的数据项数

```
    scanf ("%d    %d", &a, &b);
```

```
    printf ("a = %d, b = %d\n", a, b);
```

```
}
```

问题：如果用户输入的是123a，那么结果又会如何呢？

123a✓

a = 123, b = 121



输入数据的格式控制

```
#include <stdio.h>

main()
{
    int a, b;

    scanf("%d , %d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

格式字符串中的
普通字符原样输入

12,34✓

问题：当要求程序输出结果为

$a = 12, b = 34$

时，用户应该如何输入数据？



输入数据的格式控制

```
#include <stdio.h>
main()
{
    int a, b;

    scanf("a=%d, b=%d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

格式字符串中的
普通字符原样输入

a=12, b=34 ✓

问题：当要求程序输出结果为

a = 12, b = 34

时，用户应该如何输入数据？



scanf()的格式字符

%d	输入十进制int型
%f, %e	输入float型（不能指定输入数据的精度）
%c	输入一个字符（包括空白字符）

* 常见错误

- * `scanf ("%7.2f", &a);`
- * `scanf ("%f", a);`
- * `scanf ("%f\n", &a);`

scanf()的格式修饰符

m	输入数据的 位宽
l	加d、u前输入 long 型 加f、e前输入 double 型
L	加f、e前输入 long double 型
h	加d前输入 short 型

%ld	输入 long int 型
%hd	输入 short int 型
%f	输入 float 型
%lf	输入 double 型

为什么用**%lf**读入
双精度实数，而用
%f输出呢？



格式符%f与%lf

- 调用scanf() 输入数据时，通过地址指向变量
 - * %f告诉编译器scanf() 在该地址存储一个float型数
 - * %lf告诉编译器scanf() 在该地址存储一个double型数
- 调用printf() 输出数据时
 - * 编译器将float参数自动转换为double类型
 - * printf() 无法区分float和double

scanf()的格式修饰符

m	输入数据的位宽
l	加d、u前输入long型 加f、e前输入double型
L	加f、e前输入long double型
h	加d前输入short型
*	输入项在读入后不赋给相应的变量

scanf()的格式修饰符

```
#include <stdio.h>
main()
{
    int a, b; 跳过一个输入项

    scanf("%2d%*2d%2d", &a, &b);
    printf("a=%d, b=%d, a+b=%d\n", a, b, a+b);
}
```

问题：输入123456时程序的输出结果是什么？

123456/

a=12, b=56, a+b = 68

讨论

```
#include <stdio.h>

main()
{
    int a, b;

    scanf("%d %d", &a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

问题：修改这个程序，使用户可以以任意字符（回车、空格、制表符、逗号、其他）作为分隔符输入数据



讨论

```
include <stdio.h>
in()

int    a;
char   b;
float  c;
printf("Please input an integer:");
scanf("%d", &a);
printf("integer:%d\n", a);
printf("Please input a character:");
scanf("%c", &b);
printf("character:%c\n", b);
printf("Please input a float number:");
scanf("%f", &c);
printf("float:%f\n", c);
```

为什么得不到下面希望得到的运行结果

```
Please input an integer: 12✓
integer:12
Please input an character : a✓
character :a
Please input a float number: 3.5
float number:3.500000
```

