

# 第3章 键盘输入与屏幕输出

——用%c格式符输入数据时存在的问题

---

# 本节要讨论的主要问题

---

- 用scanf()函数的%c格式符输入字符型数据时要注意什么问题？



# 一个例子

## 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:3.50000
```

```
#include <stdio.h>  
main()  
{  
    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);  
}
```

# 一个例子

## 实际得到的运行结果：

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

用%c格式读入字符时，  
空格和回车等空白字符  
都会被当作有效字符读入



```
#include <stdio.h>  
main()  
{  
    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);  
}
```

# C格式符的问题及解决

## 程序修改后得到的运行结果

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

```
main()
```

```
{
```

```
    int    a;
```

```
    char   b;
```

```
    float  c;
```

```
→ printf("Please input an integer
```

```
→ scanf("%d", &a);
```

```
→ printf("integer:%d\n", a);
```

```
→ printf("Please input a character:");
```

```
→ getchar();
```

```
→ scanf("%c", &b);
```

```
→ printf("character:%c\n", b);
```

```
→ printf("Please input a float number:");
```

```
→ scanf("%f", &c);
```

```
→ printf("float:%f\n", c);
```

```
}
```

C语言程序设计

Please input an integer: 12 ✓

integer:12

Please input a character: a ✓

character:a

Please input a float number: 3.5 ✓

float number:3.500000

将缓冲区中的回车符读入，避免被后面的变量作为有效字符读入

# C格式符的问题及解决

```
#include <stdio.h>
main()
{
    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);
}
```

第2种解决方案:  
在%c前加一个空格

# 应用实例：输入一个加法算式

```
#include <stdio.h>
main()
{
    int    data1, data2, sum;
    char   op;

    printf("Please enter the expression\n");
    printf("data1 + data2\n");
    scanf("%d%c%d", &data1, &op, &data2);
    printf("%d%c%d = %d\n",
           data1, op, data2, data1+data2);
}
```

第1次测试



```
Please enter the expression data1 + data2
12+3 ✓
12+3 = 15
```

# 应用实例：输入一个加法算式

```
#include <stdio.h>
main()
{
    int    data1, data2, sum;
    char   op;

    printf("Please enter the expression
           data1 + data2\n");
    scanf("%d%c%d", &data1, &op, &data2);
    printf("%d%c%d = %d\n",
           data1, op, data2, data1+data2);
}
```

第2次测试

```
Please enter the expression data1 + data2
12 + 3 ✓
12 4199288 = 4199300
```





# 应用实例：输入一个加法算式

```
#include <stdio.h>
main()
{
    int    data1, data2, sum;
    char   op;

    printf("Please enter the expression\n");
    printf("data1 + data2\n");
    scanf("%d%c%d", &data1, &op, &data2);
    printf("%d%c%d = %d\n",
           data1, op, data2, data1+data2);
}
```

第3次测试

```
Please enter the expression data1 + data2
12 3 ✓
12 3 = 15
```



# 应用实例：以任意分隔符输入加法算式

```
#include <stdio.h>
main()
{
    int    data1, data2, sum;
    char   op;

    printf("Please enter the expression\n");
    printf("data1 + data2\n");
    scanf("%d%c%d",&data1, &op, &data2);
    printf("%d%c%d = %d\n",
           data1, op, data2, data1+data2);
}
```

12+3✓

12 + 3✓

12✓

+✓

3✓

忽略用户输入的算术表达式中的空白字符

# 讨论

```
#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()  
成功读入了指定数目的  
数据项呢？

问题：如果用户输入的是123a，则

```
123a✓
a = 123, b = -858993460
```

