

第一章 顺序结构

第13课 printf和scanf函数

《信息学奥赛一本通·编程启蒙 C++版》

一、格式化输入函数 scanf

scanf 函数的功能是格式化输入任意数据列表，其一般调用格式为：

scanf (格式控制符, 地址列表)

【说明】

(1) 地址列表中给出各变量的地址，可以为变量的地址，以&开头，也可以为数组、字符串的首地址。

(2) 格式控制符由%和格式符组成，作用是将要输入的字符按指定的格式输入如%d, %c 等。

scanf 函数的格式符

格式符	说 明
d, i	用于输入十进制整数
u	以无符号十进制形式输入十进制整数
o(字母)	用于输入八进制整数
X, x	用于输入十六进制整数
c	用于输入单个字符
s	用于输入字符串（非空格开始，空格结束，字符串变量以'\0'结尾）
f	用于输入实数（小数或指数均可）
e	与 f 相同（可与 f 互换）

scanf 函数的附加格式说明符

附加格式	说 明
l (字母)	用于长整型数 (%ld、%lo、%lx) 或 double 型实数 (%lf、%le)
h	用于短整型数 (%hd、%ho、%hx)
域宽 (一个整数)	指定输入所占列宽
*	表示对应输入量不赋给一个变量

二、格式化输出函数 printf

printf 函数的功能是格式化输出任意数据列表，其一般调用格式为：

printf(格式控制符，输出列表)

printf 函数的格式符

格式符	说 明
d(或 i)	以带符号的十进制形式输出整数，正数的(+)号省略不输出
u	以无符号十进制形式输出整数
x(或 X)	以十六进制无符号形式输出整数（不输出前导符 0x）
o(字母)	以八进制无符号形式输出整数（不输出前导符数字 0）
c	输出一个字符
s	输出字符串
f	以小数形式输出单、双精度，隐含输出 6 位小数
e(或 E)	以指数形式输出单、双精度，隐含输出 6 位小数
g(或 G)	自动选用%f、%e 或%E 格式中输出宽度较小的一种使用

d (整型) 和 f (浮点) 格式符

参 数	说 明
%d	输出数字长为变量数值的实际长度
%md	输出 m 位(不足补空格,大于 m 位时按实际长度输出),cout 可以用 setw 来控制场宽
%-md	m 含义同上。左对齐输出
%ld	l(小写字母)表示输出“长整型”数据
%mld	指定长整型输出宽度 m 位,左边补空格;否则,按实际位数输出
%0md,%0mld	0(数字 0)表示位数不足 m 时补 0
%lf	以小数形式输出双精度,隐含输出 6 位小数
%.mlf	以小数形式输出双精度,指定输出 m 位小数

【例 13.1】空格分隔输出

【题目描述】

读入一个字符，一个整数，一个单精度浮点数，一个双精度浮点数，然后按顺序输出它们，并且要求在他们之间用一个空格分隔。输出浮点数时保留 6 位小数。

【输入格式】

第一行是一个字符；

第二行是一个整数；

第三行是一个单精度浮点数；

第四行是一个双精度浮点数。

【输出格式】

输出字符、整数、单精度浮点数和双精度浮点数，之间用空格分隔。

【样例输入】

a

12

2.3

3.2

【样例输出】

a 12 2.300000 3.200000

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int i;
4. float f;
5. double d;
6. char c;
7. int main(){
8.     scanf("%c%d%f%lf",&c,&i,&f,&d);
9.     printf("%c %d %.6f %.6lf",c,i,f,d);
10.    return 0;
11.}
```

【例 13.2】 电子表

【题目描述】

电子表上的时间显示方法形如 `xx:xx:xx`，现在给出一个时间，单位是秒，要求按照电子表格式输出？输出保证不会超过 24 小时

【输入格式】

输入一个时间

【输出格式】

把时间转化成电子表格式输出

【样例输入】

3701

【样例输出】

01:01:41

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x;
4. int main()
5. {
6.     scanf("%d",&x);
7.     printf("%02d:%02d:%02d",x/3600,(x%3600)/60,x%60);
8.     return 0;
9. }
```

第 7 行代码 `%02d:%02d:%02d` 中 `%02d` 表示输出宽度 2 位的整数，如果位数不够 2 位，用 ‘0’ 去补。除了占位符以外所有的字符都将原封不动的输出，如这题里的 3 个 “:”。

【代码实现 2】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x;
4. int main()
5. {
6.     cin>>x;
7.     cout<<setfill('0')<<setw(2)<<x/3600<<":"
8.     <<setw(2)<<(x%3600)/60<<":"<<setw(2)<<x%3600%60;
9.     return 0;
10.}
```

`setw()` 函数用于设置字段的宽度，语法格式如下：`setw(n)`，`n` 表示宽度，用数字表示。`setw()` 函数只对紧接着的输出产生作用。当后面紧跟着的输出字段长度小于 `n` 的时候，在该字段前面用空格补齐，当输出字段长度大于 `n` 时，全部整体输出。`setw()` 默认填充的内容为空格，可以 `setfill()` 配合使用设置其他字符填充。例如第 7 行代码 `cout<<setfill('0')<<setw(2)`，设置字段宽度是 2 个，如果不够的话，用字符 '0' 进行填充。

【例 13.3】时间差

【题目描述】

给出两个时刻，问第一个时刻到第二个时刻需要经过多少秒，并保证 第一个时刻比第二个时刻早。

【输入格式】

输入有两行，第一行开始时间，第二行为结束的时间，时间都由 `xx:yy:zz` 表示，`xx` 表示小时，`yy` 表示分钟，`zz` 表示秒。

【输出格式】

只有一个整数，两个时间的差。

【样例输入】

12:34:56

12:35:00

【样例输出】

4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int a,b,c,a2,b2,c2;
4. int main()
5. {
6.     scanf("%d:%d:%d",&a2,&b2,&c2);
7.     scanf("%d:%d:%d",&a,&b,&c);
8.     printf("%d", (a*3600+b*60+c) - (a2*3600+b2*60+c2));
9. }
```

先把时间转换成最小的单位，然后在进行计算，这是我们常用的一种处理方式。

第7行代码中"`%d:%d:%d`"除了格式控制符`%d`以外还有“`:`”，这是和题目的输入要求想匹配的。

练 13.2 对齐输出

【题目描述】

读入三个整数，按每个整数占 8 个字符的宽度，右对齐输出它们，按照格式要求依次输出三个整数，之间以一个空格分开。

【输入格式】

只有一行，包含三个整数，整数之间以一个空格分开

【输出格式】

只有一行，按照格式要求依次输出三个整数，之间以一个空格分开

【样例输入】

123456789 0 -1

【样例输出】

123456789 0 -1

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x,y,z;
4. int main(){
5.     scanf("%d%d%d",&x,&y,&z);
6.     printf("%8d %8d %8d",x,y,z);
7.     return 0;
8. }
```

练 13.3 反向输出一个三位数

【题目描述】

将一个三位数反向输出，例如输入 358，反向输出 853。

【输入格式】

一个三位数 n 。

【输出格式】

反向输出 n

【样例输入】

100

【样例输出】

001

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x,y,z;
4. int main(){
5.     scanf("%c%c%c",&x,&y,&z);
6.     printf("%c%c%c",z,y,x);
7.     return 0;
8. }
```

练 13.4 糖果游戏

【题目描述】

某幼儿园里，有 5 个小朋友编号为 1、2、3、4、5，他们按自己的编号顺序围坐在一张圆桌旁。他们身上都有若干个糖果(键盘输入)，现在他们做一个分糖果游戏。从 1 号小朋友开始，将自己的糖果均分三份（如果有多余的糖果，则立即吃掉），自己留一份，其余两份分给他的相邻的两个小朋友。接着 2 号、3 号、4 号、5 号小朋友同样这么做。问一轮后，每个小朋友手上分别有多少糖果。

【输入格式】

5 个小朋友的糖果数

【输出格式】

游戏后 5 个小朋友的糖果数。（按 5 位宽度输出）

【样例输入】

8 9 10 11 12

【样例输出】

11 7 9 11 6

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int a1,a2,a3,a4,a5;
4. int main(){
5.     scanf("%d%d%d%d%d",&a1,&a2,&a3,&a4,&a5);
6.     a1/=3;a2+=a1;a5+=a1;
7.     a2/=3;a1+=a2;a3+=a2;
8.     a3/=3;a2+=a3;a4+=a3;
9.     a4/=3;a3+=a4;a5+=a4;
10.    a5/=3;a4+=a5;a1+=a5;
11.    printf("%5d%5d%5d%5d%5d",a1,a2,a3,a4,a5);
12.    return 0;
13.}
```

谢谢!

—