

# 1.

格式控制字符串中可以写多个占位符。

## 例子

```
307 {
308
309     int num1;
310     num1 = 10;
311     float f1;
312     f1 = 12.12f;
313     double d1;
314     d1 = 12.134;
315     char ch;
316     ch = 'a';
317
318
319     printf("num1变量的值是%d, f1变量的值是%f, d1变量的值是%lf, ch的值是%c", num1, f1, d1, ch);
320
321 }
```

num1变量的值是10, f1变量的值是12.120000, d1变量的值是12.134000, ch的值是aProgram ended with exit code: 0

## 2.注意:

后面的变量一定要按照顺序来写。因为在填占位符的时候是按照顺序来占位的。

否则值就会混乱

后面的变量的个数要和前面一致 如果没有变量填的就显示随机值。

## 如果少写了

```
321 int main(int argc, const char * argv[])
322 {
323
324     int num1;
325     num1 = 10;
326     float f1;
327     f1 = 12.12f;
328     double d1;
329     d1 = 12.134;
330     char ch;
331     ch = 'a';
332
333
334     printf("num1变量的值是%d, f1变量的值是%f, d1变量的值是%lf, ch的值是%c\n", num1, f1, d1);
335
336
337
338 }
```

由于d1这个变量没有占位符  
此时打印的是随机值

num1变量的值是10, f1变量的值是12.120000, d1变量的值是12.134000, ch的值是x  
Program ended with exit code: 0

如果顺序不对,类型不匹配,打印的就不正确

```
327 int num1;
328 num1 = 10;
329 float f1;
330 f1 = 12.12f;
331 double d1;
332 d1 = 12.134;
333 char ch;
334 ch = 'a';
335
336
337 printf("num1变量的值是%d, f1变量的值是%f, d1变量的值是%lf, ch的值是%c\n", ch, num1, f1, d1);
338
339
340
341
342
343 //
344 // char ch;
```

num1变量的值是97, f1变量的值是12.120000, d1变量的值是12.134000, ch的值是

Program ended with exit code: 0

Format specifies type 'double' but the argument has type 'int' 2