****在c语言中，static用来修饰变量和函数****  
1、修饰局部变量-静态局部变量  
2、修饰全局变量-静态全局变量  
3、修饰函数-静态函数

1、修饰局部变量：

*//代码1*#include <stdio.h>void test(){

int i = 0;

i++;

printf("%d ", i);}int main(){

int i = 0;

for(i=0; i<10; i++)

{

test();

}

return 0; }

*//代码2*#include <stdio.h>void test(){

*//static修饰局部变量*

static int i = 0;

i++;

printf("%d ", i);}int main(){

int i = 0;

for(i=0; i<10; i++)

{

test();

}

return 0; }

结论：  
static修饰局部变量改变了变量的生命周期，让静态局部变量出了作用域依然存在，到程序结束，生命周期才结束。

2、修饰全局变量：

*//代码1//add.c*int g\_val = 2018;*//test.c*int main(){

printf("%d\n", g\_val);

return 0; }

*//代码2//add.c*static int g\_val = 2018;*//test.c*int main(){

printf("%d\n", g\_val);

return 0; }

代码1正常，代码2在编译的时候会出现连接性错误。  
结论：  
一个全局变量被static修饰，使得这个全局变量只能在本源文件内使用，不能在其他源文件内使用。

3、修饰函数：

*//代码1//add.c*int Add(int x, int y) {

return c+y; }*//test.c*int main(){

printf("%d\n", Add(2, 3));

return 0; }

*//代码2//add.c*static int Add(int x, int y) {

return c+y; }*//test.c*int main(){

printf("%d\n", Add(2, 3));

return 0; }

代码1正常，代码2在编译的时候会出现连接性错误.  
结论：  
一个函数被static修饰，使得这个函数只能在本源文件内使用，不能在其他源文件内使用。