

C 编程中的四个调试小技巧 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/274/2021_2022_C___E7_BC_96_E7_A8_8B_E4_c67_274978.htm 1.调试标记 适用预处理

#define定义一个或多个调试标记，在代码中把调试部分使用#ifdef和#endif进行管理。当程序最终调试完成后，只需要使用#undef标记，调试代码就会消失。常用的调试标记

为DEBUG, 语句序列：#define DEBUG#ifdef DEBUG 调试代码#endif

2.运行期间调试标记 在程序运行期间打开和关闭调试标记。通过设置一个调试bool标记可以实现。这对命令行运行的程序更为方便。例如下面代码#include <iostream>

```
#include <string> using namespace std. bool debug =false. int main(int argc,char*argv[]) { for(int i=0;i < argc;i )
```

```
if(string(argv[i])== " --debug=on " ) debug = true. bool go=true. while(go) { if(debug) { 调试代码 }else {} } }
```

3.把变量和表达式转换成字符串 可是使用字符串运算符来实现转换输出定义

```
#define PR(x ) cout << "#x " = " << x << ' \n ' 4.c语言的
```

assert() 该宏在<assert>中，，当使用assert时候，给他个参数，即一个判读为真的表达式。预处理器产生测试该断言的代码，如果断言不为真，则发出一个错误信息告诉断言是什么以及它失败一会，程序会终止。#include <assert> using

```
namespace std. int main() { int i=100. assert(i!=100). //Fails } 当调试
```

```
完毕后在#include <assert> 前加入#define NDEBUG即可消除红产生的代码 }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com