如果你用过 vscode 之类的编辑器之后，
你可能会发现 DevCpp 除了安装配置方便以外。。。

可是考试还要用这个鸭。。。因为平时用的基本就是这个（codeblocks 不论）
所以我们要用一些邪门歪道魔改一下。

不过魔改的时候小心别动到其他东西把编译环境搞坏了鸭（逃

先说一个魔改之后能有效减少 zz 错误的方法---编译器警告错误开到最高级别

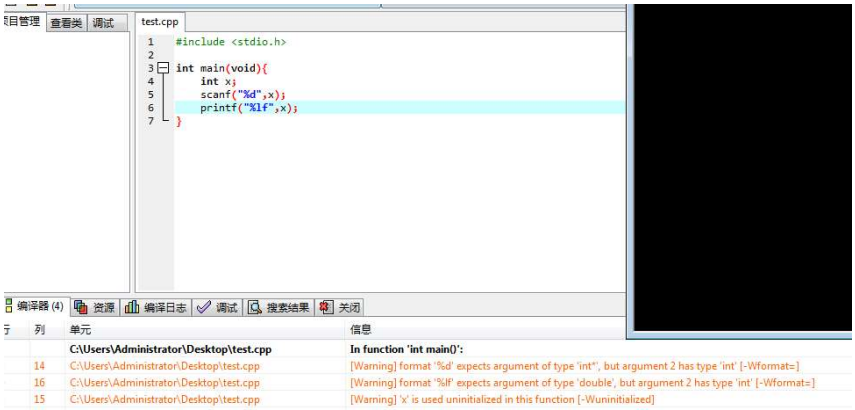
默认安装的 DevCpp 是不开启警告的（如果机房的开了那这一点可以不用看了）
对于我们现在来说，**基本上大部分**警告都是真的写错了【捂脸】。所以看到警告我们最好把他当 error 处理，仔细检查。
先上结果：
开启之前，虽说我们犯了很 zz 的错，
但是他什么也没说。。。。

```
1  #include <stdio.h>
2
3  int main(void){
4      int x;
5      scanf("%d",x);
6      printf("%lld",x);
7  }
```

编译日志 | 调试 | 搜索结果 | 关闭

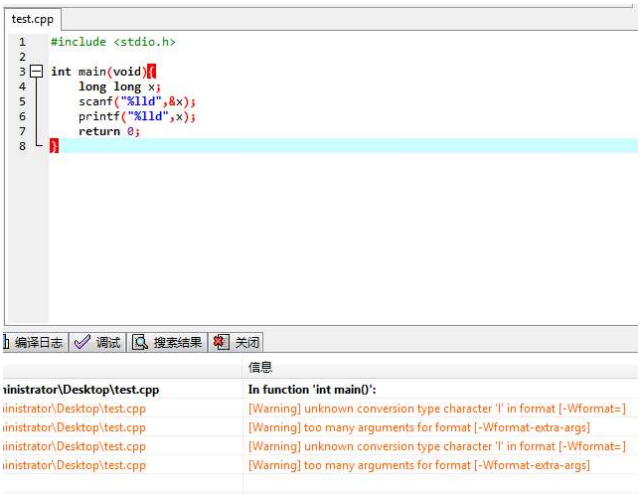
```
编译单个文件...
--------
- 文件名: C:\Users\Administrator\Desktop\test.cpp
- 编译器名: TDM-GCC 4.9.2 64-bit Release

处理 C++ 源文件...
--------
- C++ 编译器: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g
- 命令: g++.exe "C:\Users\Administrator\Desktop\test.cpp"

编译结果...
--------
- 错误: 0
- 警告: 0
- 输出文件名: C:\Users\Administrator\Desktop\test.exe
- 输出大小: 128.1015625 KiB
- 编译时间: 1.27s
```

开启之后



他告诉我们在第 5 行，scanf 格式控制符希望得到一个 int*（&x）的指针，结果我们给了它 int（x），第六行%lf 应该对应 double，结果我们给了 int，还有 x 在使用的时候没有初始化。【这样至少能在 zz 错误上少花点时间【捂脸】】

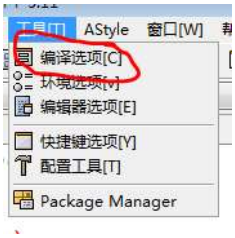
不过注意，Warning 并不一定都是错误（虽然大多数情况下是的）
比如，DevCpp 默认存为 C++，然后他就不认%lld【啊!!!!.jpg】……



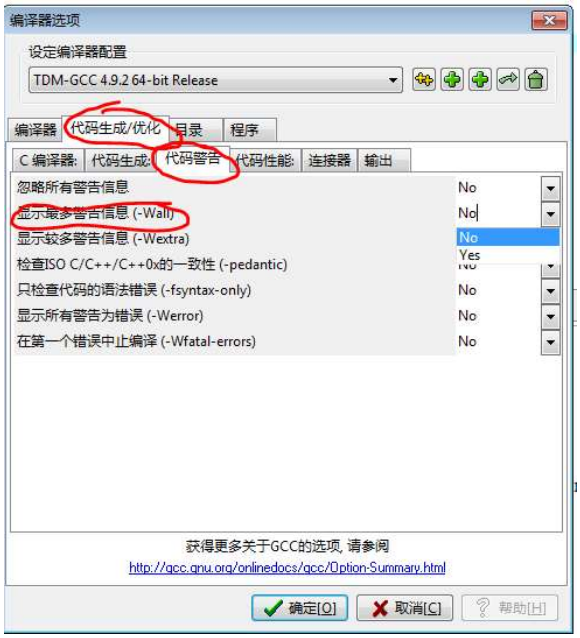
（悄悄说一句，这个警告是它不认%lld 和%llu 的结果，如果其他情况下出现了，那就要考虑是不是格式控制符和参数数量不一致了）

以及有时候不强制类型转换也有可能警告但是并没有错
所以建议大家下午熟悉一下，试着写一些能通过编译但是会显示警告的问题，然后打开编译警告测试一下会出什么样的提示。
再写一些对的代码，看看警告会误报什么
警告并不能百分百保证检测出错误，一般来说，他能检测格式控制符与类型，if（a==b）写成赋值，定义了但是却没有用到的变量，

开启 DevCpp 警告信息步骤如下。

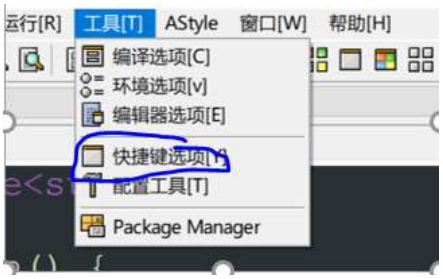


先选中编译选项，然后在代码生成/优化下面选中，显示最多警告信息，把 No 改为 Yes，保存即可



然后就是一个稍微不那么实用的小技巧了。如果你之前从来没有用过自动补全之类的东西，尝试之后又觉得不习惯的话，那就不用管了。
没错，DevCpp 也有自动补全，不过效果嘛。。。

DevCpp 的自动补全是有一套快捷键，按了之后才会显示【捂脸】，默认快捷键 Ctrl+空格正好与 windows 输入法切换冲突了【捂脸】，所以要改一下，个人习惯是改 Ctrl+Enter，大家可以根据自己喜好改。



先选快捷键选项，再拉到下面 show Code completion，然后直接再键盘上按想要的键并确定就好啦。

效果 

，按 Ctrl+回车就出来 printf 和括号了。变量名也可以。不过有时候他不认某些函数。。。）

还有一点我觉得比较重要的就是复制粘贴样例（虽说这次好像是纸质的，但是可以输入到记事本中再复制粘贴，速度会快点），详情参考助教之前发过的文件。

祝大家考个好成绩鸭~