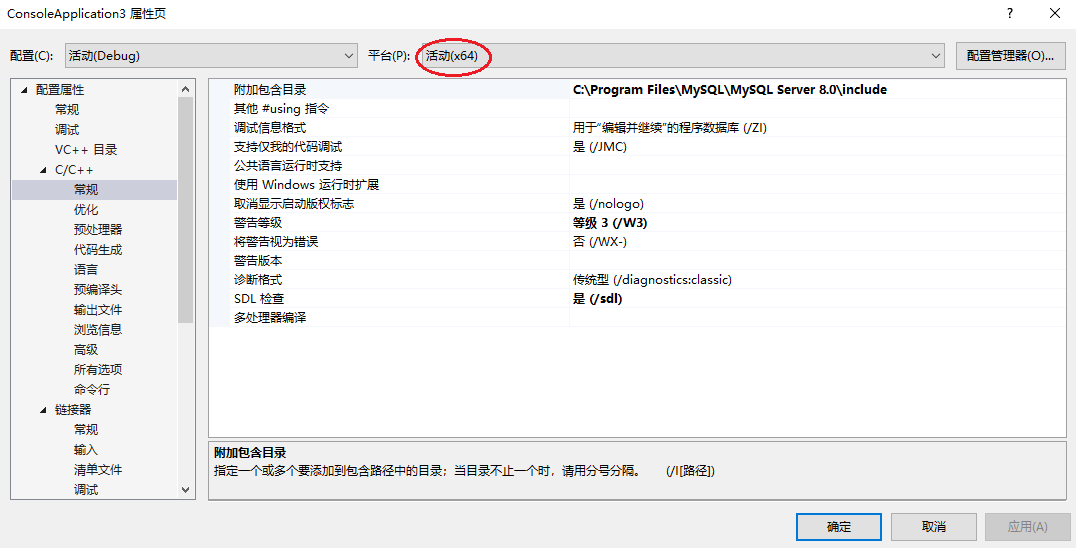
实验6指导

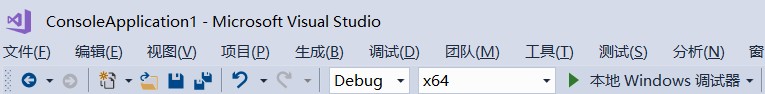
## 本次实验内容：使用嵌入式 *MySQL*，用 *C* 程序对数据库中进行创建表格和插入的操作

## 编程环境：*Visual Studio Community 2017*

1. 配置链接过程中遇到的问题：
   1. 按照教材p92-p97中*VC++.NET* 方式进行环境配置，注意打开项目属性页进行所有配置之前，要把“平台”那里改为“活动(x64)”。



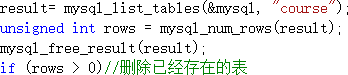
* 1. 要将*libmysql.dll*文件添加到项目文件夹中的*x64\Debug*文件夹中，这是第一次运行过程中会遇到的首要问题。
  2. （可选，一般不需要）如果以上环境手工配置不成功，可尝试在*C*语言代码中头部添加*“#pragma comment(lib,"libmysql.lib")” 。*
  3. 运行过程中需要将环境中的 *Debug* 环境变为 *x64*，如下图所示：



## 代码编写过程中遇到的问题：

* 1. 在编辑过程中发现，也许是版本问题，代码不能仅仅通过书中给出的代码进行编写，这样会导致有一些问题没办法解决。
  2. 在对表格进行检测的过程中，不能直接使用书中给出的 *mysql* 函数， 即*“mysql\_list\_tables(&mysql, "course")”*，而是添加了对于表格中行数的判

断，这样可以解决这个问题，并且 *result* 记录要被释放。



* 1. 在*create\_course\_table()*函数中发现，两个*mysql\_query()*函数连续被使用是不稳定的，可能会产生只有第一个可以查询成功，后面的查询不能成功的结果
     1. 首先尝试了网页上的解决方法，在两个 *mysql\_query()*之间使用将 *result* 的记录释放，但是这种方式在我的电脑和环境中是没有效果的

## 最后选择将创建函数中的插入初始值部分功能移动到插入函数中，这样可以解决这个问题。

* 1. 还有就是有些时候*mysql\_query()*放在*if* 条件句中进行执行也是不稳定的，所以多将其执行过程放在外面，这样的话，需要通过对函数返回值的判断来确定是否执行成功，如果执行成功其返回的是 *0*，如果不成功则返回 *1*，示例如下：



## C代码模板参考：

#include "pch.h"

#include<mysql.h>

#include<stdio.h>

#include<stdlib.h>

#include<winsock.h>

#include<iostream>

using namespace std;

MYSQL mysql;//声明为全局变量，待会在主函数，功能函数中都能对它访问

#pragma warning(disable:4996)

MYSQL\_RES \*result;

int create\_course\_table();

int insert\_rows\_into\_course\_table();

int main(int argc, char \*\*argv, char \*\*envp)

{

mysql\_init(&mysql);//获得或初始化一个MYSQL结构

//中间部分自己写

mysql\_close(&mysql);//访问完毕，关闭mysql

result = mysql\_store\_result(&mysql);

mysql\_free\_result(result);

system("pause");

return 0;

## }