测试一个.exe文件 调用 动态链接.dll文件的solution.

同时注意：solution只是把一些项目放在一起，而这些项目是完全独立的，一个项目就是一个整体，最后均可生成.dll或者是.exe文件

步骤：

1、先新建了一个空项目"test"，用于生成.exe文件

2、然后在solution中又添加了一个空项目testdll，在该项目上右键，在常规--配置类型中，把这个项目改为生成动态库.dll

3、至此，我们在一个solution中有两个项目，一个生成.exe，一个生成.dll，并且.exe调用.dll文件。项目test调用项目testdll生成的.dll

4、在testdll项目中只需要**一个文件**aa.c

在该文件中写的**函数如果想被其它项目调用**，需要加一句：

extern "C" int \_\_declspec(dllexport)add(int x,int y);

这一句表示int add(int x,int y)函数要被其它项目调用。而

\_\_declspec(dllexport)可以看做关键字。

然后，在aa.c文件下写这个函数的定义就好了，写完之后生成.dll文件

5、在test项目中也只需要**一个文件**bb.c

在该文件中想要用testdll项目中的add函数，首先需要找到这个项目.dll文件；然后需要找到该文件中包含的add函数。具体如下：

**第一步：找到testdll项目生成的.dll文件**：

HINSTANCE hDll; //函数句柄

hDll=LoadLibrary(\_\_TEXT("D:\\study software\\VS 2005 vc++8.0\\Had installed\\VC\\projects\\testdll\\Debug\\programdll.dll"));

通过LoadLibrary函数，找到这个programdll.dll文件的地址，并把它赋给hDll。参数使用了绝对地址，并且用了\_\_TEXT( )，并且绝对地址之间的分隔符是“\\”，而不是“\”。

**第二步：从这个.dll文件中找到add函数**

typedef int(\*lpAddFun)(int,int); //函数指针定义别名:lpAddFun

lpAddFun addFun; //通过这个函数指针类型，定义一个函数指针。

addFun=(lpAddFun)GetProcAddress(hDll,"add");//通过GetProcAddress函数，并以.dll地址，和要寻找的函数add作为参数，把返回结果赋给addFun.

6、如果一切都顺利，就已经结束了。接下来我们就可以通过函数指针:addFun找到int add(int x,int y)函数的入口地址。使用方法直接为int i=addFun(3,5);

addFun就相当于add。

7、最后记得释放这个动态库：FreeLibrary(hDll);