C与C++混合编程大致有两种情况：

1. 在C++中调用C语言的函数

例：程序C\_C++

a.h

#ifndef A\_H

#define A\_H

int sum(int , int);

#endif // A\_H

a.c

#include "a.h"

int sum(int a, int b)

{

return (a + b);

}

b.h

#ifndef B\_H

#define B\_H

// cpp文件会默认定义\_\_cplusplus

// 之所以采用两个条件指令#ifdef

// 是因为如果不是cpp文件，则直接

// 定义#include "a.h"

#ifdef \_\_cplusplus

// extern "C"是提示编译器

// a.h中的函数按照C语言的方式编译

// C++中支持函数重载，而C语言不支持

// 这导致在编译时，函数被内部命名的方式

// 不同

extern "C"

{

#endif

#include "a.h"

#ifdef \_\_cplusplus

};

#endif

void func();

#endif // B\_H

b.cpp

#include "b.h"

#include <iostream>

using std::cout;

using std::endl;

void func()

{

int a = 2;

int b = 3;

cout << "sum = " << sum(a, b) << endl; // 调用C语言的函数

}

test.cpp

// 在C++中调用C语言的函数

#include <iostream>

#include "b.h"

using std::cout;

using std::endl;

int main(int argc, char\* argv[])

{

func();

return 0;

}

例：程序C\_C++\_2

1. 在C语言中调用C++函数

a.h

#ifndef A\_H

#define A\_H

#ifdef \_\_cplusplus

extern "C" // 按照C语言的方式编译

{

#endif

int sum(int a, int b);

#ifdef \_\_cplusplus

};

#endif

#endif // A\_H

a.cpp

#include "a.h"

int sum(int a, int b)

{

return (a + b);

}

b.h

#ifndef B\_H

#define B\_H

void func();

#endif

b.c

#include "b.h"

#include <stdio.h>

#include "a.h"

void func()

{

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

}

test.c

// 在C语言中调用C++函数

#include "b.h"

int main(int argc, char\* argv[])

{

func();

return 0;

}