**C++第二次实验报告**

1. **设计程序的思路**

本次实验程序用到了C++中的内联函数，在函数说明前冠以关键字“inline”，该函数就被声明为内联函数，又称为内置函数。每当程序中出现对该函数的调用时，C++编译器使用函数体中的代码插入到调用该函数的语句处，同时用实参取代形参，以便在程序运行时不再进行函数调用。本程序中内联函数为:

inline void attack(int&attack\_num)

{

cout<<"攻击数量"<<endl;

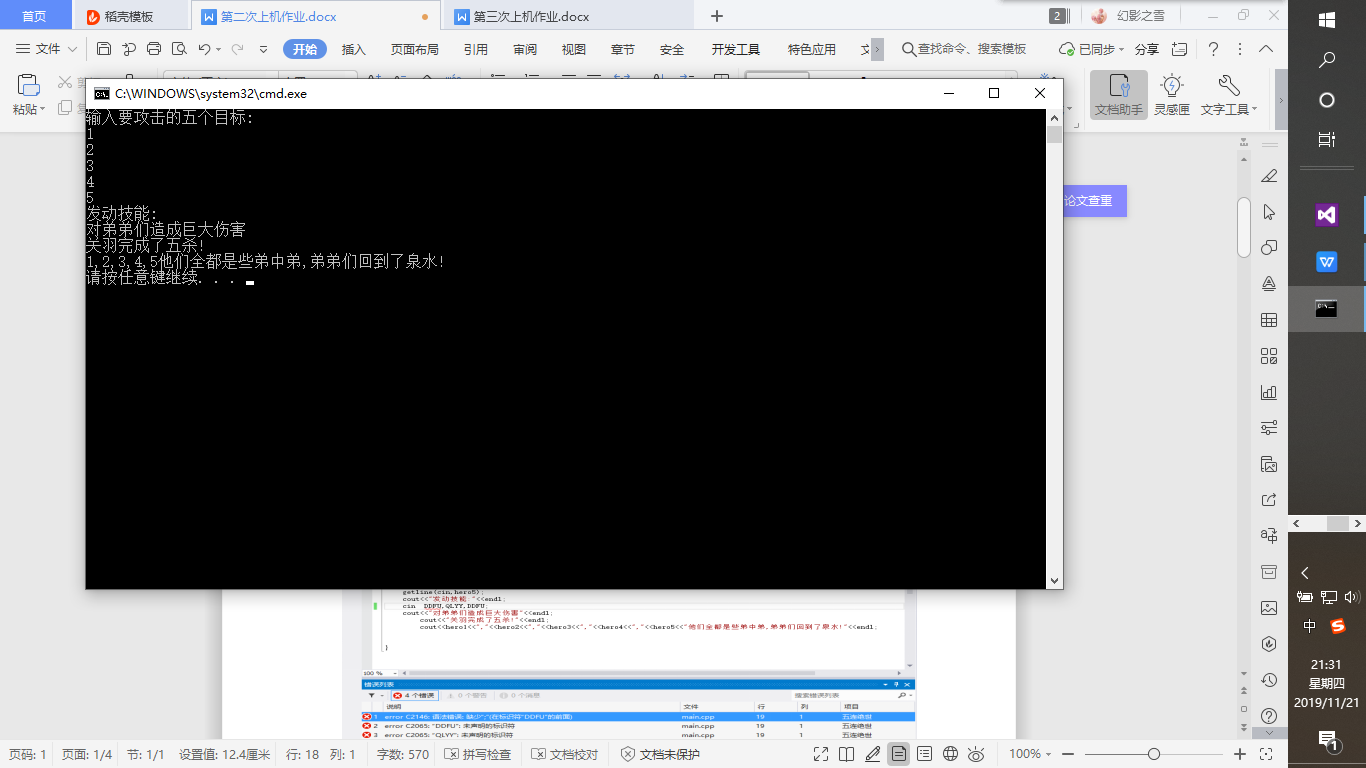
}

在main.cpp中使用内联函数令main.cpp调试的同时引用skills.cpp和skills.h，从而得到五连绝世的结果。

1. **程序执行时遇到的错误**

本次实验所用到的程序因要用到内联函数所以相比于第一次实验来说较为复杂，

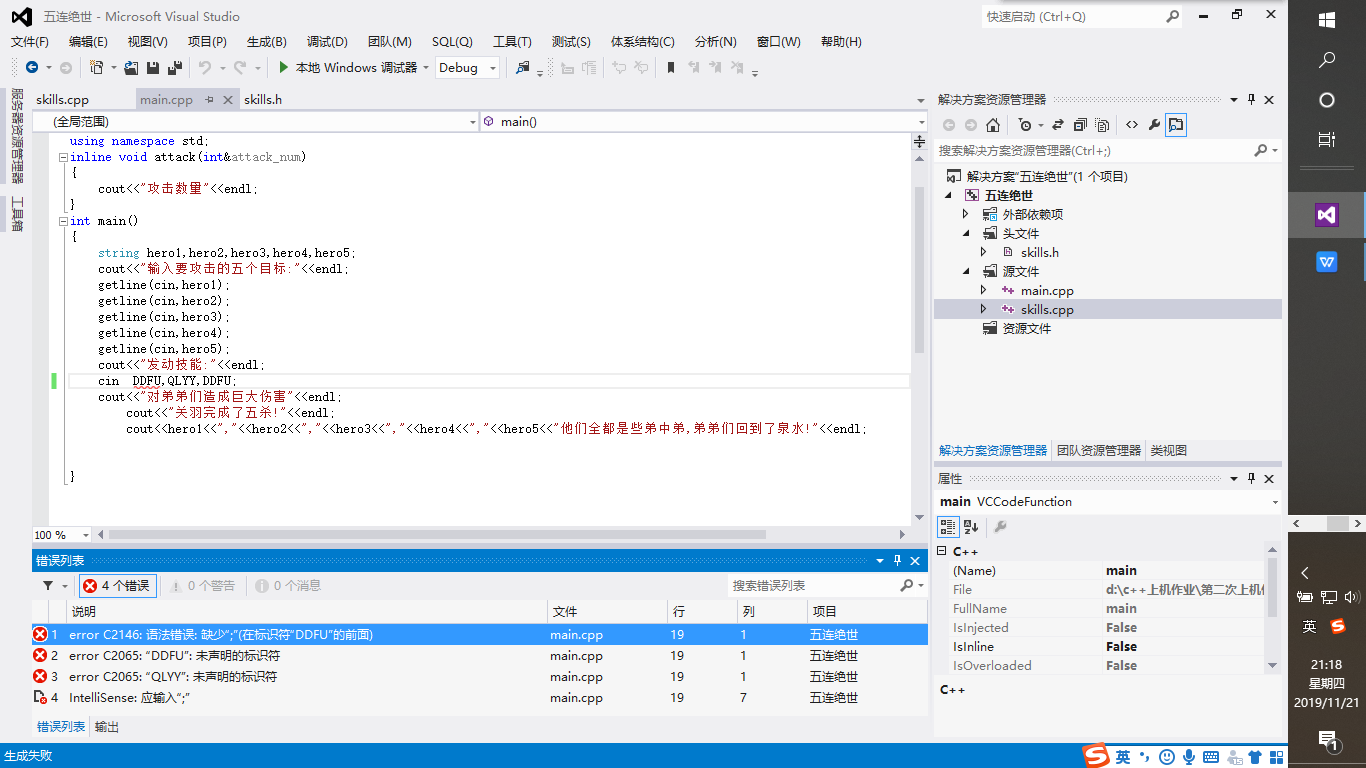
该错误是在程序没错误的情况下输出结果无符合设想结果，输出结果如下图所示：



本实验是要在输出结果中敲入技能名称时可以达到关羽对敌人释放技能进而达成五连绝世的效果，该结果无法输入技能名称，很显然该结果没能达到这个预期。

接下来就是对输入技能名称这段程序的修改，这次修改就造成了第二个错误。

错误程序如下图所示：



在这里只想用cin将技能名称用键盘输入，从而达到当在输出结果中敲入技能名称时可以达到关羽对敌人释放技能进而达成五连绝世。但在定义技能名称输入的时候出现了问题，cin的使用出现了问题，正确程序应为：

string DDFU,QLYY,DFTJ;

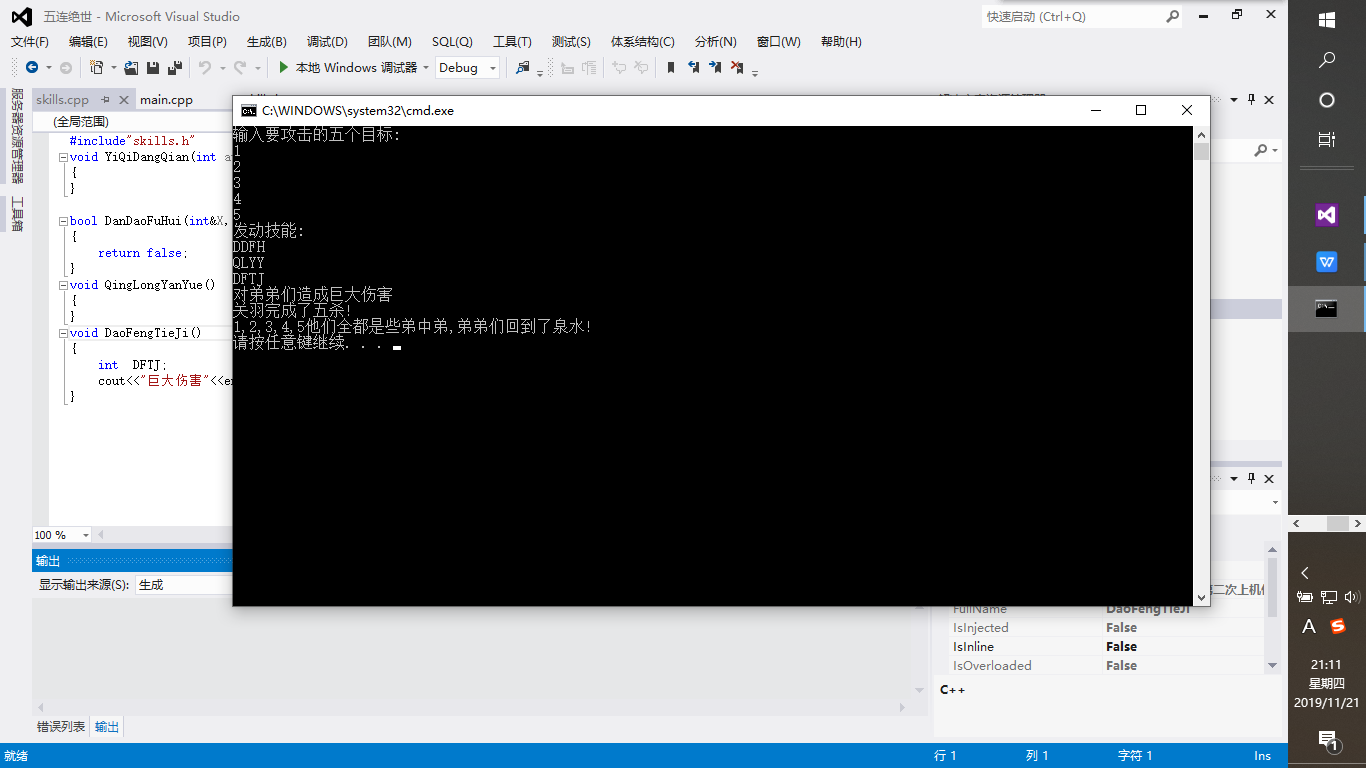
getline(cin,DDFU);

getline(cin,QLYY);

getline(cin,DFTJ);

cout<<"对弟弟们造成巨大伤害"<<endl;

正确程序的输出结果如图所示：



1. **程序的主体代码**

**skills.cpp**

#include"skills.h"

void YiQiDangQian(int attacks, bool&Skill3Enable)

{

}

bool DanDaoFuHui(int&X, int&Y, int num)

{

return false;

}

void QingLongYanYue()

{

}

void DaoFengTieJi()

{

int DFTJ;

cout<<"巨大伤害"<<endl;

}

**main.cpp**

#include<iostream>

#include"skills.h"

#include<string>

using namespace std;

inline void attack(int&attack\_num)

{

cout<<"攻击数量"<<endl;

}

int main()

{

string hero1,hero2,hero3,hero4,hero5;

cout<<"输入要攻击的五个目标:"<<endl;

getline(cin,hero1);

getline(cin,hero2);

getline(cin,hero3);

getline(cin,hero4);

getline(cin,hero5);

cout<<"发动技能:"<<endl;

string DDFU,QLYY,DFTJ;

getline(cin,DDFU);

getline(cin,QLYY);

getline(cin,DFTJ);

cout<<"对弟弟们造成巨大伤害"<<endl;

cout<<"关羽完成了五杀!"<<endl;

cout<<hero1<<","<<hero2<<","<<hero3<<","<<hero4<<","<<hero5<<"他们全都是些弟中弟,弟弟们回到了泉水!"<<endl;

}

**skills.h**

#include<iostream>

using namespace std;

void YiQiDangQian(int attacks, bool&Skill3Enable);

bool DanDaoFuHui(int &X, int &Y, int num);

void QingLongYanYue();

void DaoFengTieJi(int DFTJ );