



§. 基础知识题 - cin与cout的基本使用

要求:

- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**9月22日前**网上提交本次作业（在“文档作业”中提交）



§. 基础知识题 - cin与cout的基本使用

贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图

```
Microsoft Visual Studio 调试控制台
Hello, world!
D:\Workspace\VS2019-Demo\Debug\cpp-demo.exe (进程 7484)已退出，代码为 0。
按任意键关闭此窗口...
```

例：有效贴图

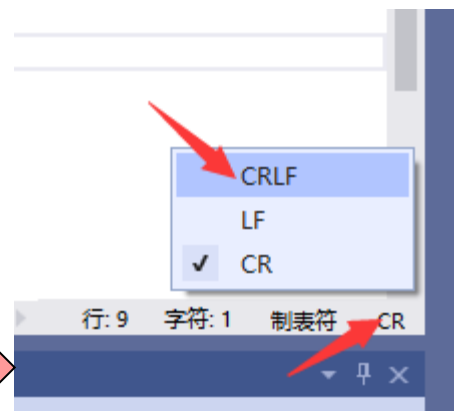
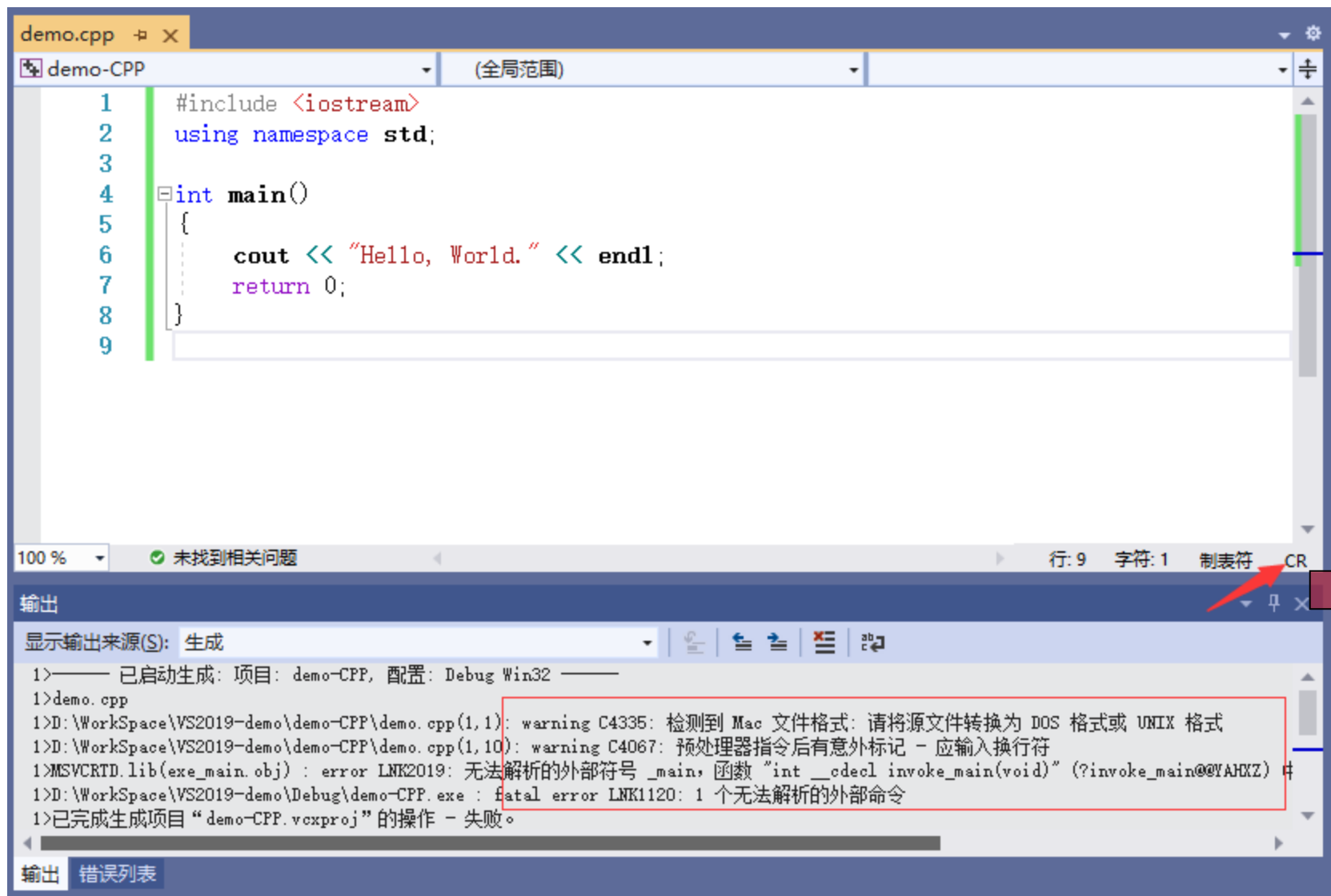
```
Microsoft Visual Studio 调试控制台
Hello, world!
```



§. 基础知识题 - cin与cout的基本使用

附：用WPS等其他第三方软件打开PPT，将代码复制到VS2022中后，如果出现类似下面的**编译报错**，则观察源程序编辑窗

的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可





特别提示:

- 1、做题过程中，先按要求输入，如果想替换数据，也要先做完指定输入
- 2、如果替换数据后出现某些问题，先记录下来，不要问，等全部完成后，还想不通再问(也许你的问题在后面的题目中有答案)
- 3、不要偷懒、不要自以为是的脑补结论!!!
- 4、先得到题目要求的小结论，再综合考虑上下题目间关系，得到综合结论
- 5、这些结论，是让你记住的，不是让你完成作业后就忘掉了
- 6、换位思考(从老师角度出发)，这些题的目的是希望掌握什么学习方法？



§ . 基础知识题 - cin与cout的基本使用

基本知识点:

- 1、cin是按格式读入，到空格、回车、非法为止
- 2、cin的输入必须以回车结束，输入的内容放在输入缓冲区中，从输入缓冲区去取得所需要的内容后，多余的内容还放在输入缓冲区中，等待下次读入（如果程序结束，则操作系统会清空输入缓冲区）
- 3、系统会自动根据cin后变量的类型按**最长原则**来读取合理数据
- 4、变量读取后，系统会判断输入数据是否超过变量的范围，若超过则**置内部的错误标记**并返回一个**不可信**的值（不同编译器处理不同）
 - 4.1、cin输入完成后，通过cin.good()/cin.fail()可判断本次输入是否正确
 - 4.2、cin碰到非法字符后会置错误标记位，后面会一直错（**如何恢复还未学到，先放着**）
 - 4.3、cin连续输入多个int时，碰到非法字符，下一个是0，再下面才是随机值
 - 4.4、cin超范围后，不同类型的数据处理不同，如果细节记不清，问题不大，但一定要知道有这回事，别奇怪
 - 4.5、cin超范围和赋值超范围是不同的

5、cout根据数据类型决定输出形式

输入	cin.good() 返回	cin.fail() 返回
正确范围 +回车/空格/非法输入	1	0
错误范围 +回车/空格/非法输入	0	1
非法输入	0	1

6、先认真看课件!!!



§. 基础知识题 - cin与cout的基本使用

1、cout的基本理解

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

int main()
{
    /* 第1组 */
    cout << "This is a C++ program." << endl;

    /* 第2组 */
    cout << "This is " << "a C++ " << "program." << endl;

    /* 第3组 */
    cout << "This is "
         << "a C++ "
         << "program."
         << endl;

    /* 第4组 */
    cout << "This is ";
    cout << "a C++ ";
    cout << "program.";
    cout << endl;

    return 0;
}
```

Microsoft Visual Studio 调试控制台

```
This is a C++ program.
This is a C++ program.
This is a C++ program.
This is a C++ program.
```

第3组和第4组在语句上的区别是：

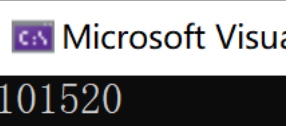


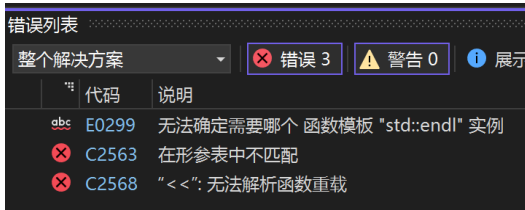
第3组整体是一个语句，而第4组是由四个语句组成



§. 基础知识题 - cin与cout的基本使用

1、cout的基本理解

B. 观察下列4个程序的运行结果，回答问题并将各程序的运行结果截图贴上(如果有错则贴错误信息截图)

<pre>#include <iostream> using namespace std; int main() { int a=10, b=15, c=20; cout << a << b << c; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int a=10, b=15, c=20; cout << a, b, c; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int a=10, b=15, c=20; cout << (a, b, c) << endl; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int a=10, b=15, c=20; cout << a, b, c << endl; return 0; }</pre> 
<p>解释这3个程序输出不同的原因： 第1个程序依次输出a、b、c三个变量，且中间没有符号间隔；第2个程序相当于是一个逗号表达式，其中第1项是cout<<a；第3个程序相当于输出逗号表达式的值c。</p>			<p>解释错误原因：cout和endl分别属于逗号表达式的第一项和第三项</p>
<p>结论：一个流插入运算符 << 只能输出__1__个数据.</p>			



§. 基础知识题 - cin与cout的基本使用

1、cout的基本理解

C. 观察下列2个程序的运行结果，回答问题并将各程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
int main()
{
    char ch = 65;
    cout << ch << endl;
    return 0;
}
```



```
#include <iostream>
using namespace std;
int main()
{
    int ch = 65;
    cout << ch << endl;
    return 0;
}
```



解释这两个程序输出不同的原因：

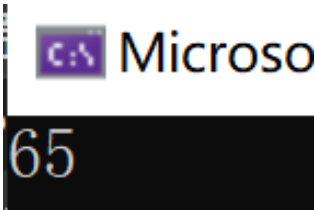

cout会根据数据类型自动调整输出结果，如第一个程序中ch属于char类，所以会输出65对应的ASCII码，而第二个程序中ch属于int类，所以会输出数字65.



§ . 基础知识题 - cin与cout的基本使用

1、cout的基本理解

D. 程序同C，将修改后符合要求的程序及运行结果贴上

<pre>#include <iostream> using namespace std; int main() { char ch = 65; cout << (int) ch << endl; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int ch = 65; cout << (char) ch << endl; return 0; }</pre> 
在char类型不变的情况下，要求输出为65 (不允许添加其它变量)	在int类型不变的情况下，要求输出为A (不允许添加其它变量)



§. 基础知识题 - cin与cout的基本使用

1、cout的基本理解

E. 程序同C，将修改后符合要求的程序及运行结果贴上

```
#include <iostream>
using namespace std;
int main()
{
    char ch = 65;
    cout << ch + 0 << endl;
    return 0;
}
```



在char类型不变的情况下，要求输出为65
(不允许添加其它变量，
不允许使用任何方式的强制类型转换)



§. 基础知识题 - cin与cout的基本使用

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

A. 运行下面的程序，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

<pre>#include <iostream> using namespace std; int main() { short k; cin >> k; cout << "k=" << k << endl; return 0; }</pre>	<div>1、输入：123✓（✓代表回车键，下同）</div> <div>2、输入：123 456✓（一个空格）</div> <div>3、输入：123 456✓（多个空格）</div> <div>4、输入：123m✓</div> <div>5、输入：m✓</div> <div>6、输入：123✓（持续多个空格后，再输入123，按回车）</div> <div>7、输入：123✓（持续多个空格后，按回车） 123✓（再输入123，按回车）</div> <div>8、输入：✓ ... ✓ 123✓（持续多个空回车后，输入123）</div> <div>分析结果： 1、在前面有正确输入的情况下，回车、空格、（对int型而言是非法的字符）m的作用是？ 终止输入 2、直接输入若干空格和回车后，再输入正确，变量是否能得到正确的值？ 可以 3、直接输入（对int型而言是）非法的数据m，输出是？ 0</div>
<p>基础知识：</p> <p>short的最小值是：_-32768_</p> <p>short的最大值是：__32767__</p>	



§ . 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

B. 运行下面的程序，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

<pre>#include <iostream> using namespace std; int main() { short k; cin >> k; cout << "k=" << k << endl; cout << "cin.good()=" << cin.good() << endl; cout << "cin.fail()=" << cin.fail() << endl; return 0; }</pre>	<p>贴图即可，不需要写分析结果</p> <p>1、输入：123✓ （正确+回车）</p> <p>2、输入：123 456✓ （正确+空格）</p> <p>3、输入：-123m✓ （正确+非法字符）</p> <p>4、输入：m✓ （直接非法字符）</p> <p>5、输入：54321✓ （超上限）</p> <p>6、输入：-40000✓ （超下限）</p>
<p>结论：</p> <p>多个输入中，编号__4、5、6__输入的k值是可信的</p>	<p>本题要求VS+Dev</p>



1、

```
Microsoft Visual Studio  
123  
k=123  
cin.good()=1  
cin.fail()=0
```

```
\\Mac\Home\De  
123  
k=123  
cin.good()=1  
cin.fail()=0
```

4、

```
Microsoft Visual Studio  
m  
k=0  
cin.good()=0  
cin.fail()=1
```

```
\\Mac\Home\De  
m  
k=0  
cin.good()=0  
cin.fail()=1
```

2、

```
Microsoft Visual Studio  
123 456  
k=123  
cin.good()=1  
cin.fail()=0
```

```
\\Mac\Home\Desktop  
123 456  
k=123  
cin.good()=1  
cin.fail()=0
```

5、

```
Microsoft Visual Studio  
54321  
k=32767  
cin.good()=0  
cin.fail()=1
```

```
\\Mac\Home\Desktop  
54321  
k=32767  
cin.good()=0  
cin.fail()=1
```

3、

```
Microsoft Visual Studio  
-123m  
k=-123  
cin.good()=1  
cin.fail()=0
```

```
\\Mac\Home\Desktop  
-123m  
k=-123  
cin.good()=1  
cin.fail()=0
```

6、

```
Microsoft Visual Studio  
-40000  
k=-32768  
cin.good()=0  
cin.fail()=1
```

```
\\Mac\Home\Desktop  
-40000  
k=-32768  
cin.good()=0  
cin.fail()=1
```



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

B-Compare. 运行下面的**对比**程序（cin输入与赋值），观察运行结果并与B的输出结果进行对比分析

```
#include <iostream>
using namespace std;
int main()
{
    short k1, k2, k3, k4, k5;

    k1 = 12345;
    k2 = 54321;
    k3 = 70000;
    k4 = -12345;
    k5 = -54321;

    cout << k1 << endl;
    cout << k2 << endl;
    cout << k3 << endl;
    cout << k4 << endl;
    cout << k5 << endl;

    return 0;
}
```

Microsoft

```
12345
-11215
4464
-12345
11215
```

B的输入:

1、输入: 12345✓（合理范围）

对应本例的k1=12345

2、输入: 54321✓（超上限但未超同类型的u_short上限）

对应本例的k2=-11215

3、输入: 70000✓（超上限且超过同类型的u_short上限）

对应本例的k3=4464

4、输入: -12345✓（合理范围）

对应本例的k4=-12345

5、输入: -54321✓（超下限）

对应本例的k5=11215

对比分析:

1. B中cin输入超上限时，变量接受32767，超下限时，变量接受-32768；

2. B-Compare中将超上限值但未超同类型的u short上限，将补码赋给变量，若超过u short上限，则会被截断。

u_short=unsigned short



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

C. 仿B，自行构造不同测试数据，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可

```
#include <iostream>
using namespace std;

int main()
{
    int k;
    cin >> k;
    cout << "k=" << k << endl;
    cout << "cin.good()=" << cin.good() << endl;
    cout << "cin.fail()=" << cin.fail() << endl;
    return 0;
}
```

结论：
多个输入中，编号_2、3、5____输入的k值是可信的

贴图即可，不需要写分析结果

1、输入：_1024_✓ （合理范围）

Microsoft Visual

```
1024
k=1024
cin.good()=1
cin.fail()=0
```

signed int

2、输入：_4000000000_✓ （超限）

选择 Microsoft V

\\Mac\Home\Desk

```
4000000000
k=2147483647
cin.good()=0
cin.fail()=1
```

```
4000000000
k=32767
cin.good()=0
cin.fail()=1
```

3、输入：_20000000000_✓ （u_int上限）

Microsoft Visual S

\\Mac\Home\D

```
20000000000
k=2147483647
cin.good()=0
cin.fail()=1
```

```
20000000000
k=32767
cin.good()=0
cin.fail()=1
```

4、输入：_-1024_✓ （合理范围）

Microsoft Visu

```
-1024
k=-1024
cin.good()=1
cin.fail()=0
```

5、输入：_-4000000000_✓ （超限）

Microsoft Visual

\\Mac\Home\D

```
-4000000000
k=-2147483648
cin.good()=0
cin.fail()=1
```

```
-4000000000
k=-32768
cin.good()=0
cin.fail()=1
```

本题要求VS+Dev



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

C-Compare. 仿B-Compare, 构造**对比**程序 (cin输入与赋值, int型), 观察运行结果并与C的输出结果进行对比分析

注: 具体对比程序及输出结果等不要再贴图, 自行完成即可

需要回答下列问题 (回答问题不是完成作业, 而是自己真的弄懂了概念后的总结) :

1、输入/赋值超int上限但未超同类型的u_int上限, 两者是否一致? 如果有区别, 区别是?

不一致, 输入很好说明, 超过int上限, 则该变量的值为 $2^{31}-1$;

赋值则是将二进制信息赋给变量。

以4000000000为例, 为u_int型 $4000000000 > 2^{31}-1$

将4000000000转成二进制: 11101110 01101011 00101000 00000000

将该组二进制赋值给变量, 同时C++将其解释为变量的补码, 所以变量对应十进制为-294967296



2、输入/赋值超int上限且超同类型的u_int上限，两者是否一致？如果有区别，区别是？

不一致，输入很好说明，超过int上限，则该变量的值为 $2^{31}-1$ ；

赋值则是将二进制信息赋给变量

以8000000000为例，为u_int型 $8000000000 > 232$

将8000000000转成二进制： 1 11011100 11010110 01010000 00000000

将该组二进制赋值给int型变量时，会出现截断现象，int只有32位，所以最高位1会被舍弃，即C++解释器将输出补码11011100 11010110 01010000 00000000对应的十进制-589934592

选择6000000000时，得到的十进制数是正数，原因是舍去最高位1之后，符号位为0，所以为正数。

将6000000000转成二进制： 1 01100101 10100000 10111100 00000000

```
#include <iostream>
using namespace std;

int main()
{
    int a = 8000000000;
    cout << a << endl;
    return 0;
}
```

Microsoft Visual Studio

-589934592



3、输入/赋值超int下限，两者是否一致？如果有区别，区别是？

不一致，输入很好说明，超过int下限，则该变量的值为 -2^{32} ；
赋值则是将二进制信息赋给变量。

以-4000000000为例，为long long型 $-4000000000 < 2^{32}$

将-4000000000转成二进制： 10000000 00000000 00000000 00000000 11101110 01101011

00101000 00000000

反码： 11111111 11111111 11111111 11111111 00010001 10010100 11010111 11111111

补码： 11111111 11111111 11111111 11111111 00010001 10010100 11011000 00000000

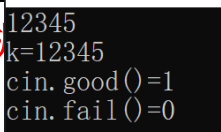
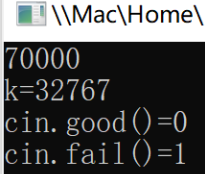
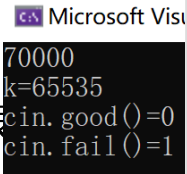
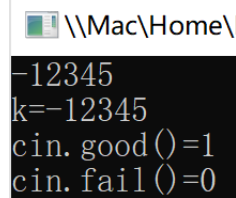
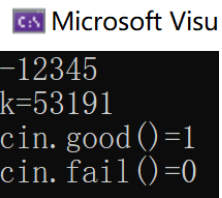
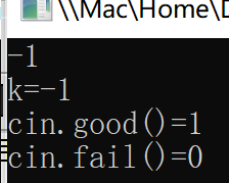
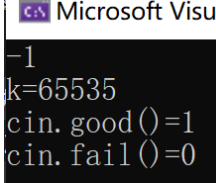
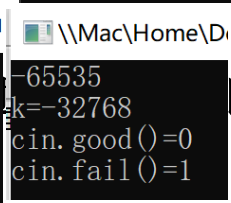
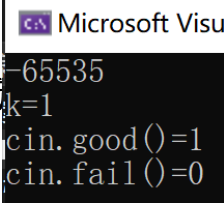
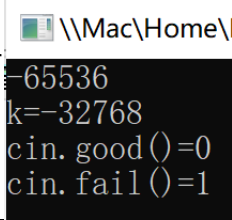
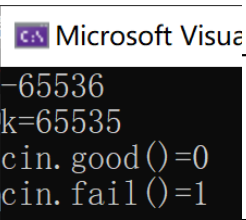
而int只有四个字节，所以会发生高位截断，保留00010001 10010100 11011000 00000000，C++
解释器将其视作一个数的补码，对应十进制数294967296



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

D. 运行下面的程序，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

<pre>#include <iostream> using namespace std; int main() { unsigned short k; cin >> k; cout << "k=" << k << endl; cout << "cin.good()=" << cin.good() << endl; cout << "cin.fail()=" << cin.fail() << endl; return 0; }</pre>	<div>贴图即可，不需要写分析</div> <div>1、输入：12345✓（合理范围） </div> <div>2、输入：70000✓（超上限） </div> <div>3、输入：-12345✓（负数但未超过short下限） </div> <div>4、输入：-1✓（负数且未超过short下限） </div> <div>5、输入：-65535✓（负数且接近加负号后的下限） </div> <div>6、输入：-65536✓（负数且超过加负号后的下限） </div>
<p>结论：</p> <p>多个输入中，编号_2、3、4、5、6_输入的k值是可信的</p>	<div>题要求VS+Dev</div>

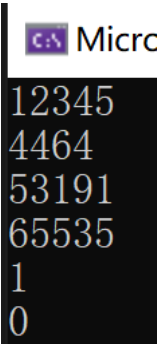


§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

D-Compare. 仿B-Compare构造的对比程序（cin输入与赋值，u_short型），观察运行结果并与D的输出结果进行对比分析

<pre>#include <iostream> using namespace std; int main() { u_short k1, k2, k3, k4, k5, k6; k1 = 12345; k2 = 70000; k3 = -12345; k4 = -1; k5 = -65535; k6 = -65536; cout << k1 << endl; cout << k2 << endl; cout << k3 << endl; cout << k4 << endl; cout << k5 << endl; cout << k6 << endl; return 0; }</pre>	<p>贴图即可，不需要写分析结果</p> <p>1、输入：12345✓ （合理范围） 对应本例的k1=12345</p> <p>2、输入：70000✓ （超上限） 对应本例的k2=4464</p> <p>3、输入：-12345✓ （负数但未超过short下限） 对应本例的k3=53191</p> <p>4、输入：-1✓ （负数且未超过short下限） 对应本例的k4=65535</p> <p>5、输入：-65535✓ （负数且未超过u_short上限加负号后的下限） 对应本例的k5=1</p> <p>6、输入：-65536✓ （负数且超过u_short上限加负号后的下限） 对应本例的k6=0</p>	<p>u_short=unsigned short</p>
		<p>本题要求VS+Dev</p>





§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

E. 仿D，自行构造不同测试数据，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

<pre>#include <iostream> using namespace std; int main() { unsigned int k; cin >> k; cout << "k=" << k << endl; cout << "cin.good()=" << cin.good() << endl; cout << "cin.fail()=" << cin.fail() << endl; return 0; }</pre>	<p>贴图即可，不需要写分析结果</p> <p>1、输入：_1024_ ✓ （合理范围）</p> <p>2、输入：_ 8000000000 _ ✓ （超过上限）</p> <p>3、输入：_-1024_ ✓ （负数但未超过下限）</p> <p>4、输入：_ -4000000000 _ ✓ （负数且超过下限）</p> <p>5、输入：_-8000000000_ ✓ （负数且超过负号后的下限）</p>
<p>结论：</p> <p>多个输入中，编号__2、3、4、5__输入的k值是可信的</p>	<p>int=unsigned int</p> <p>本题要求VS+Dev</p>



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

E-Compare. 仿B-Compare, 构造**对比**程序 (cin输入与赋值, u_int型), 观察运行结果并与E的输出结果进行对比分析

注: 具体对比程序及输出结果等不要再贴图, 自行完成即可

需要回答下列问题 (回答问题不是完成作业, 而是自己真的弄懂了概念后的总结) :

1、输入/赋值超u_int上限, 两者是否一致? 如果有区别, 区别是?

不一致, 输入很好说明, 超过int上限, 则该变量的值为 $2^{31}-1$;

赋值则是将二进制信息赋给变量

以8000000000为例, 为u_int型 $8000000000 > 2^{32}$

将8000000000转成二进制: 1 11011100 11010110 01010000 00000000

将该组二进制赋值给int型变量时, 会出现截断现象, int只有32位, 所以最高位1会被舍弃,

即C++解释器将输出补码11011100 11010110 01010000 00000000对应的十进制-589934592

选择6000000000时, 得到的十进制数是正数, 原因是舍去最高位1之后, 符号位为0, 所以为正数。

将6000000000转成二进制: 1 01100101 10100000 10111100 00000000



2、输入/赋值为负数但未超int下限，两者是否一致？如果有区别，区别是？

一致

3、输入/赋值为负数且未超过u_int上限加负号后的下限，两者是否一致？如果有区别，区别是？

不一致，输入按照补码传递；

4、输入/赋值为负数负数且超过u_int上限加负号后的下限？如果有区别，区别是？

不一致，输入很好说明，超过int下限，则该变量的值为 -2^{32} ；



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

B-E. 总结

名词解释:

输入正确 - 指数学上合法的数，但不代表一定在C/C++的某类型数据的数据范围内（下同）

综合2.B~2.E，给出下列问题的分析及结论：

- 1、signed数据在输入正确且范围合理的情况下 原样输出
- 2、signed数据在输入正确但超上限（未超同类型unsigned上限）的情况下 按该类型最大值输出
- 3、signed数据在输入正确且超上限（超过同类型unsigned上限）的情况下 按该类型最大值输出
- 4、signed数据在输入正确但超下限范围的情况下 按该类型最小值输出
- 5、unsigned数据在输入正确且范围合理的情况下 原样输出
- 6、unsigned数据在输入正确且超上限的情况下 按unsigned型最大值输出
- 7、unsigned数据在输入正确但为负数（未超同类型signed下限）的情况下 按补码形式输出
- 8、unsigned数据在输入正确且为负数（超过同类型signed下限）的情况下 按补码形式输出
- 9、unsigned数据在输入正确且为负数（超过同类型unsigned上限加负号后的下限）的情况下 按unsigned型最大值输出

对比：cin输入与变量赋值，在输入/右值超范围的情况下，表现是否相同？总结规律 输入取最值，赋值取补码
cin输入与变量赋值，在输入/右值合理范围的情况下，表现是否相同？总结规律 相同



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

F. 运行下面的程序，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cin >> ch;

    cout << "ch=" << int(ch) << endl;
    cout << "ch=" << ch << endl;

    return 0;
}
```

1、键盘输入A（单个图形字符）

```
A
ch=65
ch=A
```

2、键盘输入\b（退格键的转义符）

```
\b
ch=92
ch=
```

3、键盘输入\101（A的ASCII码的8进制转义表示）

```
\101
ch=92
ch=
```

4、键盘输入\x41（A的ASCII码的16进制转义表示）

```
\x41
ch=92
ch=
```

5、键盘输入65（A的ASCII码的十进制整数形式表示）

```
65
ch=54
ch=6
```

6、键盘输入Ctrl+C（注意：是Ctrl+C组合键，要有输入法栏）

```
ch=
```

7、键盘输入Ctrl+z（注意：是Ctrl+z组合键，注意不要有输入法栏）

```
^Z
ch=-52
ch=
```



§. 基础知识题 - cin与cout的基本使用

2、cin的基本理解 - 单数据情况

G. 运行下面的程序，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    float f;
    cin >> f;

    cout << f << endl;
    cout << setprecision(20) << f << endl;

    return 0;
}
```

Microsoft Visual Studio 调试

```
1. 23456e2
123. 456
123. 45600128173828125
```

Microsoft Visual Studio 调试

```
-1. 23456e2
-123. 456
-123. 45600128173828125
```

Microsoft Visual Studio 调试

```
6. 7e38
0
0
```

Microsoft Visual Studio 调试

```
-2. 3e39
0
0
```

//注：20已超float和double精度

Microsoft Visual Studio 调试控制台

```
-1. 23e-30
-1. 23e-30
-1. 2299999549998595325e-30
```

1、键盘输入123. 456 （合理范围正数，小数形式）

Microsoft Visual Studio 调试

```
123. 456
123. 456
123. 45600128173828125
```

2、键盘输入1. 23456e2 （合理范围正数，指数形式）

Microsoft Visual Studio 调试

```
-123. 456
-123. 456
-123. 45600128173828125
```

3、键盘输入-123. 456 （合理范围负数，小数形式）

4、键盘输入-1. 23456e2 （合理范围负数，指数形式）

Microsoft Visual Studio 调试

```
123. 456789
123. 457
123. 456787109375
```

5、键盘输入123. 456789 （合理范围，但超有效位数）

6、键盘输入6. 7e38 （超上限但数量级未超，仍是 10^{38} ）

Microsoft Visual Studio 调试

```
1. 7e39
0
0
```

7、键盘输入1. 7e39 （超上限且数量级已超 10^{38} ）

8、键盘输入-2. 3e39 （超上限且数量级已超 10^{38} ）

Microsoft Visual Studio 调试控制台

```
1. 23e-30
1. 23e-30
1. 2299999549998595325e-30
```

9、键盘输入1. 23e-30 （合理范围整数但指数很小）

10、键盘输入-1. 23e-30 （合理范围负数但指数很小）



§. 基础知识题 - cin与cout的基本使用

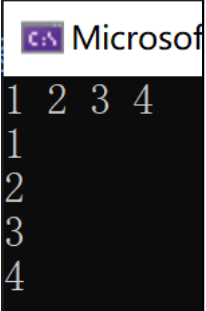
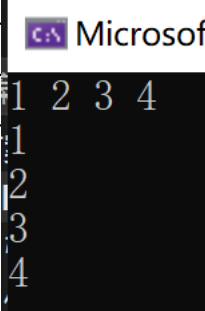
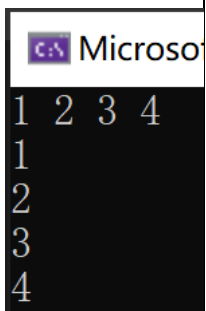
此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - cin与cout的基本使用

3、cin的基本理解 - 多个同类型数据的情况

A. 观察下列3个程序的运行结果，回答问题并将各程序的运行结果截图贴上(如果有错则贴错误信息截图)

<pre>#include <iostream> using namespace std; int main() { int a, b, c, d; cin >> a >> b >> c >> d; cout << a << endl; cout << b << endl; cout << c << endl; cout << d << endl; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int a, b, c, d; cin >> a >> b >> c >> d; cout << a << endl; cout << b << endl; cout << c << endl; cout << d << endl; return 0; }</pre> 	<pre>#include <iostream> using namespace std; int main() { int a, b, c, d; cin >> a; cin >> b; cin >> c; cin >> d; cout << a << endl; cout << b << endl; cout << c << endl; cout << d << endl; return 0; }</pre> 
---	---	--

1、程序运行后，输入：1 2 3 4✓，观察输出结果

2、解释第2个和第3个程序的cin语句的使用区别：

第2个程序只有一个cin语句，和第1个程序相同；而第3个程序有四个cin语句，通过空格间隔，依次将缓存区中的数据输入给变量



§. 基础知识题 - cin与cout的基本使用

3、cin的基本理解 - 多个同类型数据的情况

B. 程序同A，观察不同输入下的运行结果（贴图在清晰可辨的情况下，能小）

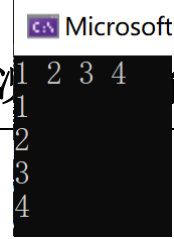
```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;

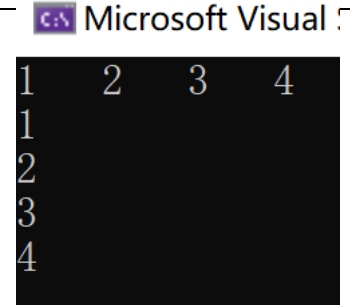
    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    cout << d << endl;

    return 0;
}
```

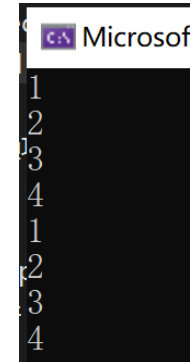
1、输入：1 2 3 4✓



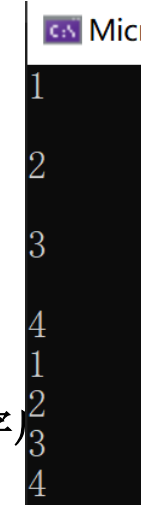
2、输入：1 2 3 4✓（每个数字间多于一个空格）



3、输入：1✓
2✓
3✓
4✓（每个数字后立即加回车）



4、输入：1✓
2✓
3✓
4✓（每个数字后加回车 + 多个空回车）



结论：在输入正确的情况下，回车和空格的作用？ 终止对一个变量的输入



§. 基础知识题 - cin与cout的基本使用

3、cin的基本理解 - 多个同类型数据的情况

C. 程序同A，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

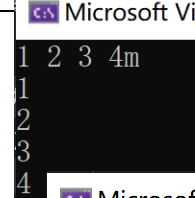
```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;

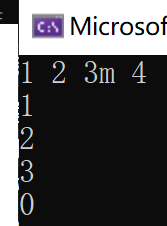
    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    cout << d << endl;

    return 0;
}
```

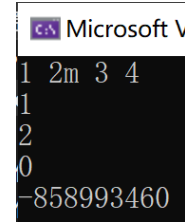
1、输入：1 2 3 4m✓



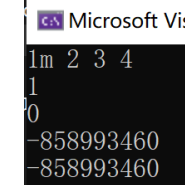
2、输入：1 2 3m 4✓



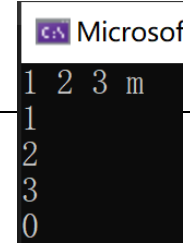
3、输入：1 2m 3 4✓



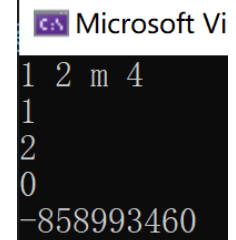
4、输入：1m 2 3 4✓



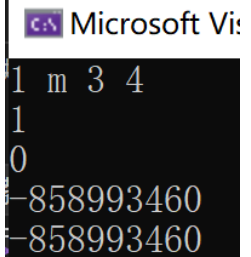
5、输入：1 2 3 m✓



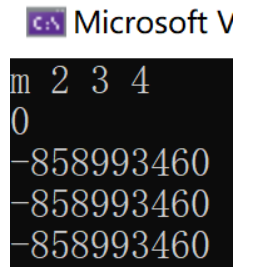
6、输入：1 2 m 4✓



7、输入：1 m 3 4✓



8、输入：m 2 3 4✓





总结：多个cin输入时，错误输入出现在不同位置对输入正确性的影响

要求：综合观察运行结果，加上自己的思考，给出总结性的结论，这个结论要能对多个输入情况下不同位置的错误情况有普遍适应性，而不仅仅是简单的根据结论说错在1/2/3/4位置

（提示：从什么位置开始值不可信？）

1. 错误输入在最后时，对输入无影响；
2. 错误输入在中间时，分两种情况：直接代替正确输入，则此时该位对应的变量输入为0，它后面的位置对应的值均不可信；若接在正确输入后面，则此时该位对应的变量输入正确，它后面的位置对应的值均不可信；

（不可信的值默认是-858993460）



§. 基础知识题 - cin与cout的基本使用

3、cin的基本理解 - 多个同类型数据的情况

D. 观察不同输入下的运行结果（贴图在清晰可辨的情况下）

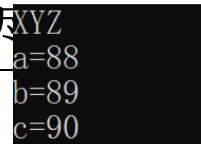
```
#include <iostream>
using namespace std;

int main()
{
    char a, b, c;
    cin >> a >> b >> c;

    cout << "a=" << int(a) << endl;
    cout << "b=" << int(b) << endl;
    cout << "c=" << int(c) << endl;

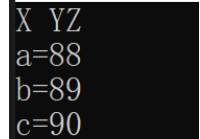
    return 0;
}
```

1、输入：XYZ✓



XYZ
a=88
b=89
c=90

2、输入：X YZ✓



X YZ
a=88
b=89
c=90

3、输入：Ctrl+C✓ （表示按Ctrl+C组合键，注意不要有输入法栏，下同）



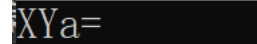
a=

4、输入：X Ctrl+C✓



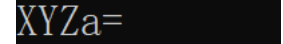
Xa=

5、输入：XY Ctrl+C✓



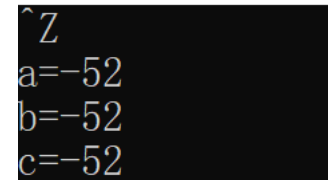
XYa=

6、输入：XYZ Ctrl+C✓



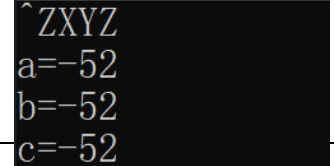
XYZa=

7、输入：Ctrl+z✓ （若未出结果则继续输入，可以按回车后多行输入，打印后观察结果）



^Z
a=-52
b=-52
c=-52

8、输入：Ctrl+zXYZ✓ （若未出结果则继续输入，可以按回车后多行输入，打印后观察结果）



^ZXYZ
a=-52
b=-52
c=-52



总结：多个cin输入时char型数据时

1、能否输入空格

不能

2、Ctrl+C在输入中表示什么？（可自行查阅资料，若资料与表现不符，信哪个？）

中断程序

3、Ctrl+z在输入中表示什么？（可自行查阅资料，若资料与表现不符，信哪个？）

结束输入

4、Ctrl+z后不按回车而继续输入的其它字符，能否被读入？

不能



§. 基础知识题 - cin与cout的基本使用

3、cin的基本理解 - 多个同类型数据的情况

E. 自行构造测试数据，观察不同输入下的运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    float a, b, c;
    cin >> a >> b >> c;

    cout << "a=" << a << endl;
    cout << setprecision(20) << a << endl;

    cout << "b=" << b << endl;
    cout << setprecision(20) << b << endl;

    cout << "c=" << c << endl;
    cout << setprecision(20) << c << endl;

    return 0;
}
```

1、输入: `_1e50 1 1`_____✓ （第1个超上限，2/3正常）

2、输入: `_-1e50 1 1`_____✓ （第1个超下限，2/3正常）

3、输入: `_1 1e50 _1`_____✓ （1/3正常，第2个超上限）

4、输入: `_1 -1e50 1`_____✓ （1/3正常，第2个超下限）

5、输入: `_1 1 1e50`_____✓ （1/2正常，第3个超上限）

6、输入: `_1 1 -1e50`_____✓ （1/2正常，第3个超下限）

总结:

1、多个cin输入时，错误输入出现在不同位置对输入正确性的影响

要求: 综合观察运行结果，加上自己的思考，给出总结性的结论，这个结论要能对多个输入情况下不同位置的错误情况有普遍适应性，而不仅仅是简单的根据结论说错在1/2/3位置

（提示: 从什么位置开始值不可信？）

错误输入对前面正确输入无影响，该输入对应的变量为赋值为0，它后面的变量（若有）均被赋值为-107374176

2、将float替换为double，上述结论是否仍然成立？ 成立



1.

```
Microsoft Visual Studio
1e50 1 1
a=0
0
b=-107374176
-107374176
c=-107374176
-107374176
```

2.

```
Microsoft Visual Studio
-1e50 1 1
a=0
0
b=-107374176
-107374176
c=-107374176
-107374176
```

3.

```
Microsoft Visual Studio
1 1e50 1
a=1
1
b=0
0
c=-107374176
-107374176
```

4.

```
Microsoft Visual Studio
1 -1e50 1
a=1
1
b=0
0
c=-107374176
-107374176
```

5.

```
Microsoft Visual Studio
1 1 1e50
a=1
1
b=1
1
c=0
0
```

6.

```
Microsoft Visual Studio
1 1 -1e50
a=1
1
b=1
1
c=0
0
```



§. 基础知识题 - cin与cout的基本使用

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - cin与cout的基本使用

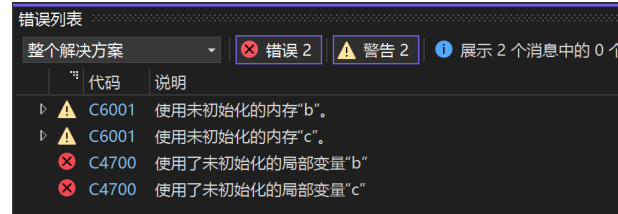
4、cin的基本理解 - 其他情况

A. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

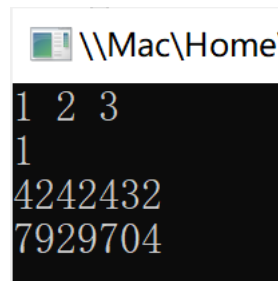
```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cin >> a,b,c;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    return 0;
}
```

1、如果编译有error或warning，则贴相应信息的截图



2、如果能运行(包括有warning)，则输入三个正确的int型数据
(例 :1 2 3✓)，观察输出



Vs中运行报错，Dev中可以运行

3、分析为什么只有某个变量的结果是正确的

cin >> a, b, c; 相当于一个逗号表达式，cin >> a 属于其中一部分，而b, c则是单独的，由因为b, c在之前未被定义，所以在vs中报错

本题要求VS+Dev



§. 基础知识题 - cin与cout的基本使用

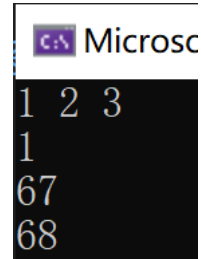
4、cin的基本理解 - 其他情况

B. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
using namespace std;
int main()
{
    int a=66, b=67, c=68;
    cin >> a,b,c;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    return 0;
}
```

1、运行后，输入三个正确的int型数据(例 :1 2 3✓，注意不要是预置值)，观察输出



2、通过观察三个变量的输出，你得到了什么结论？

`cin >> a, b, c;`

对这一语句，当我们输入三个数据时，cin只能将第一个数据输入给变量a，而b，c仅作为逗号表达式的一部分，没有实际作用。



§ . 基础知识题 - cin与cout的基本使用

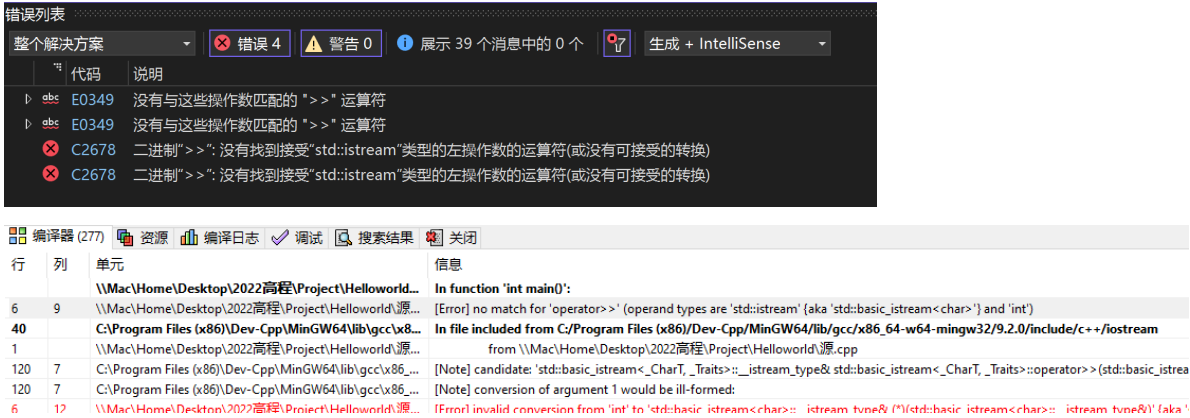
4、cin的基本理解 - 其他情况

C. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> 5;
    cin >> a+10;

    cout << a << endl;
    return 0;
}
```

1、如果编译有error或warning，则贴相应信息的截图(信息太多则前五五行)



2、分析为什么编译有错

```
cin >> 5; // cin后面是常量
cin >> a + 10; // cin后面是表达式
```

3、结论：流提取运算符后面必须跟_b____，不能是_a、c_

- a) 常量 b) 变量 c) 表达式

本题要求VS+Dev



§. 基础知识题 - cin与cout的基本使用

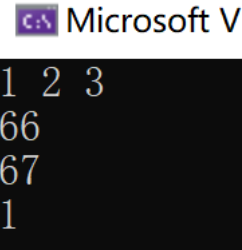
4、cin的基本理解 - 其他情况

D. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
using namespace std;
int main()
{
    int a=66, b=67, c=68;
    cin >> (a,b,c);

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    return 0;
}
```

1、运行后，输入三个正确的int型数据(例 :1 2 3✓，注意不要是预置值)，观察输出



2、通过观察三个变量的输出，你得到了什么结论？

a和b没有受cin>>(a, b, c);影响，而c的值发生了改变，且值为第1个输入1

3、和B进行比较，分析为什么结果有差异

(a, b, c) 逗号表达式的值为最右端的c，所以相当于cin >> c；而对a和b没有影响

4、和C进行比较，与C得出的结论矛盾吗？

不矛盾，逗号表达式的值为变量c



§. 基础知识题 - cin与cout的基本使用

4、cin的基本理解 - 其他情况

E. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

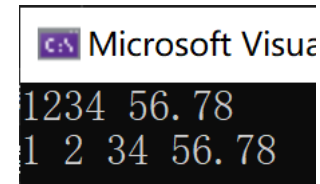
```
#include <iostream>
using namespace std;
int main()
{
    char c1, c2;
    int a;
    float b;
    cin >> c1 >> c2 >> a >> b;

    cout << c1 << ' ' << c2 << ' ' << a << ' ' << b << endl;
    return 0;
}
```

注：┐表示空格

1、输入：1234┐56.78✓

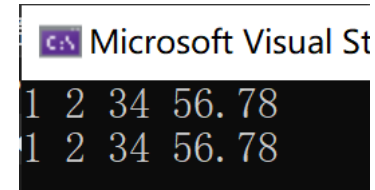
输出：



Microsoft Visual Studio
1234 56.78
1 2 34 56.78

2、输入：1┐2┐34┐56.78✓

输出：



Microsoft Visual Studio
1 2 34 56.78
1 2 34 56.78

3、分析在以上两种不同输入的情况下，为什么输出相同（提示：空格的作用）

cin根据变量类型进行读取，
空格使cin切换到下一个变量进行输入



§. 基础知识题 - cin与cout的基本使用

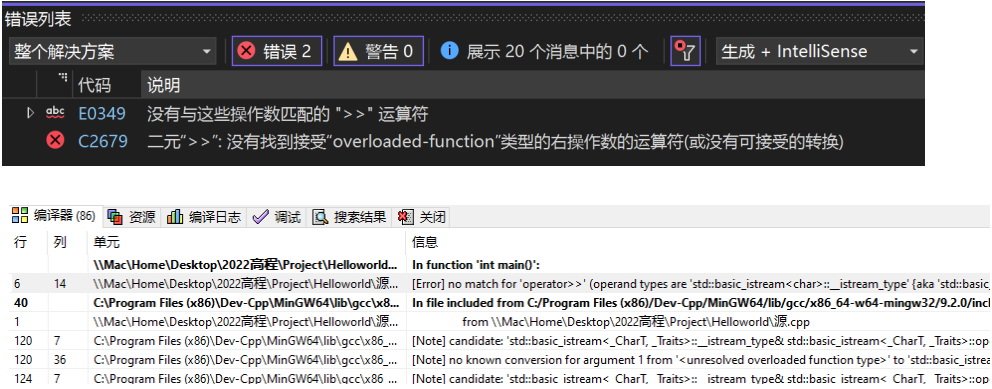
4、cin的基本理解 - 其他情况

F. 程序如下，观察编译及运行结果（贴图在清晰可辨的情况下尽可能小）

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a >> endl;

    return 0;
}
```

1、如果编译有error或warning，则贴相应信息的截图(信息太多则前五五行)



2、结论：在cin中不能跟__endl_____

本题要求VS+Dev



§. 基础知识题 - cin与cout的基本使用

此页不要删除，也没有意义，仅仅为了分隔题目