

§. 基础知识题 – 关系运算、逻辑运算与选择结构



要求:

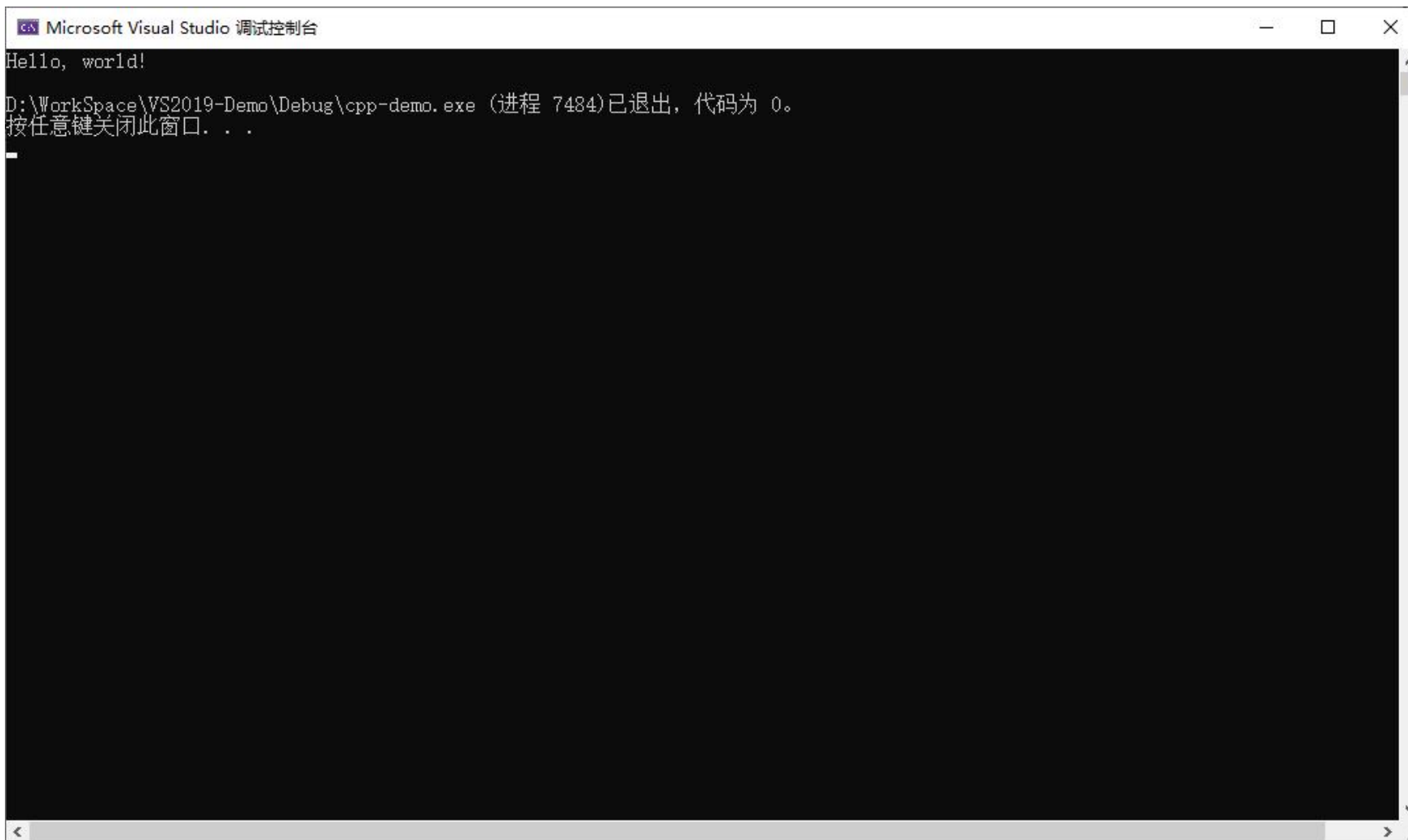
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**10月12日前**网上提交本次作业（在“文档作业”中提交）

§. 基础知识题 - 关系运算、逻辑运算与选择结构

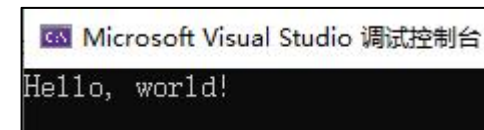


贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图

A screenshot of the Microsoft Visual Studio debug console window. The window is titled "Microsoft Visual Studio 调试控制台". It contains the following text: "Hello, world!", "D:\Workspace\VS2019-Demo\Debug\cpp-demo.exe (进程 7484)已退出, 代码为 0.", and "按任意键关闭此窗口. . .". The window is large and shows the full output of the program.

例：有效贴图

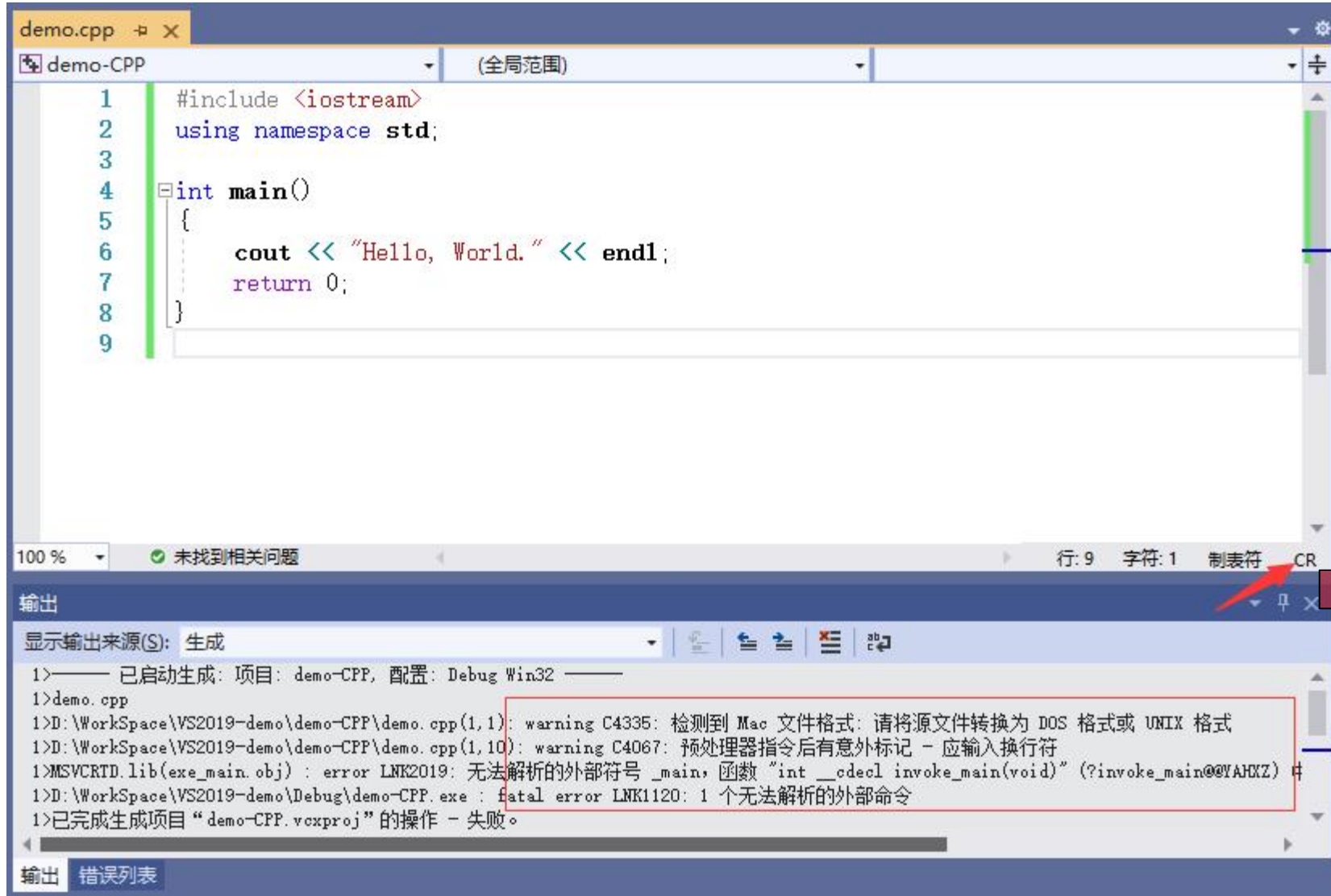
A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the "Hello, world!" output. The window title is "Microsoft Visual Studio 调试控制台".



§. 基础知识题 - 关系运算、逻辑运算与选择结构

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



§. 基础知识题 - 关系运算、逻辑运算与选择结构



特别提示:

★ 本次作业的答案，除特别提示外，课件上都有，好好看课件!!!



§. 基础知识题 - 关系运算、逻辑运算与选择结构

1、关系运算符的求值顺序

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()
{
    int a=1, b=2, c=3, d;

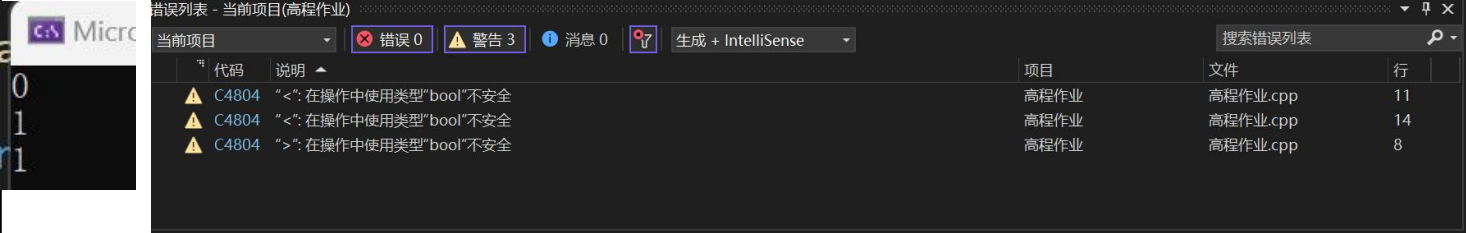
    d = a > b > c;
    cout << d << endl;

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

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

    return 0;
}
```

1、贴运行结果



2、VS下为什么会有三个warning? 说说你的理解

VS报在操作中使用类型“bool”不安全, 此代码使用“<”和“>”比较整型和布尔值, 编译器认为这个操作可能和编程的意图不符, 使用报warning。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

1、关系运算符的求值顺序

B. 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()
{
    int a=3, b=2, c=1, d;

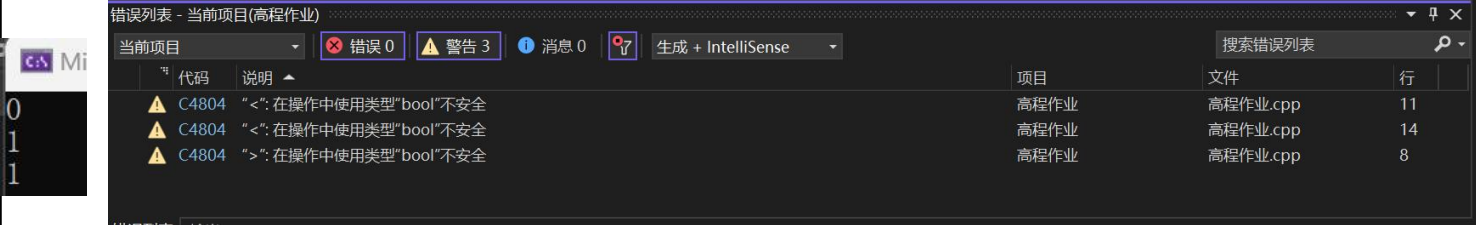
    d = a > b > c;
    cout << d << endl;

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

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

    return 0;
}
```

1、贴运行结果



2、 $a > b > c$ 这个式子，按常规理解， $3 > 2 > 1$ 是正确的，为什么结果是0？
因为实际运算是这样的：

$a > b$ 实际上为 $3 > 2 = \text{true}$ ，表达式的值转化为整型1；

$a > b > c$ 实际上为 $1 > 1 = \text{false}$ ，表达式的值转化为整型0

$a < b < c$ 这个式子，按常规理解， $3 < 2 < 1$ 是错误的，为什么结果是1？
因为实际运算是这样的：

$a < b$ 实际上为 $3 < 2 = \text{false}$ ，表达式的值转化为整型0；

$a < b < c$ 实际上为 $0 < 1 = \text{true}$ ，表达式的值转化为整型1

$b > a < c$ 这个式子，按常规理解， $2 > 3 < 1$ 是错误的，为什么结果是1？
因为实际运算是这样的：

$b > a$ 实际上为 $2 > 3 = \text{false}$ ，表达式的值转化为整型0；

$b > a < c$ 实际上为 $0 < 1 = \text{true}$ ，表达式的值转化为整型1

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

2、关系运算符与实数

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)

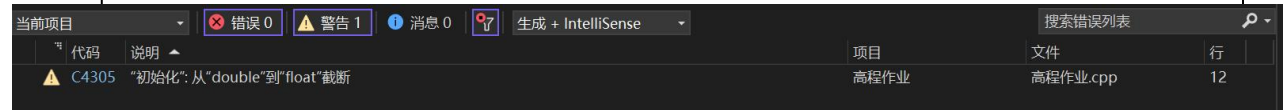
```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    float f1 = 100.25;
    cout << (f1 - 100.25) << endl;
    cout << (f1 == 100.25) << endl;
    cout << (fabs(f1-100.25) < 1e-6) << endl;

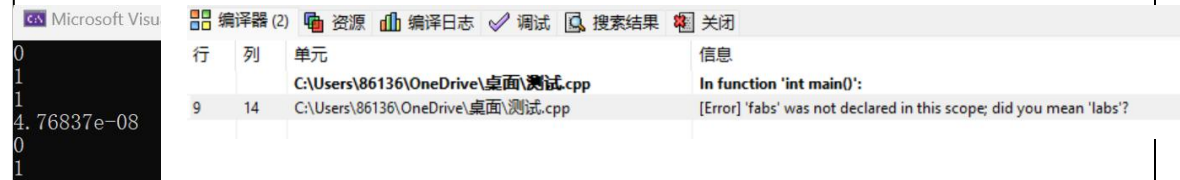
    float f2 = 1.2;
    cout << (f2 - 1.2) << endl;
    cout << (f2 == 1.2) << endl;
    cout << (fabs(f2-1.2) < 1e-6) << endl;

    return 0;
}
```

1、贴VS+Dev下的运行结果



2、删除第2行的#include<cmath>后，再次贴VS+Dev的运行结果



3、由本例得出的结论，实数进行相等比较时的通用方法是 两个数相差的绝对值如果小于一定值，则会认为相等；对于float型，这个差值为1e-6



§. 基础知识题 - 关系运算、逻辑运算与选择结构

2、关系运算符与实数

B. 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)

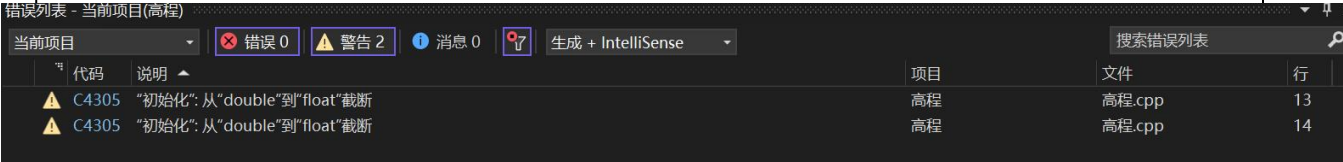
```
#include <iostream>
#include <cmath> //VS可不加
using namespace std;

int main()
{
    double d1=123.456789012345678;
    double d2=123.456789123456789;
    cout << (d1==d2) << endl;
    cout << (fabs(d1-d2)<1e-6) << endl;
    cout << (fabs(d1-d2)<1e-7) << endl;

    float f1=123.456789012345678;
    float f2=123.456789123456789;
    cout << (f1==f2) << endl;
    cout << (fabs(f1-f2)<1e-6) << endl;
    cout << (fabs(f1-f2)<1e-7) << endl;

    return 0;
} //VS有两个warning
```

1、贴运行结果



2、观察fabs(**)<1e-6 和 fabs(**)<1e-7在float和double下的表现，哪个相同？哪个不同？为什么？
fabs(**)<1e-6相同， fabs(**)<1e-7不同。
因为double型比float型的精度位数更高, 有效位数更多, 所以面对更小位数的差值时，double型可以表示得出来而float表示不出来。

§. 基础知识题 - 关系运算、逻辑运算与选择结构



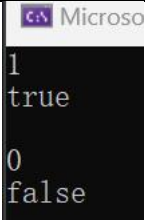
此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

3、逻辑常量与逻辑变量

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)

<pre>#include <iostream> using namespace std; int main() { cout << true << endl; cout << "true" << endl; cout << endl; cout << false << endl; cout << "false" << endl; return 0; }</pre>	<div>1、贴运行结果</div>  <div>2、解释 true 和 "true" 的区别 (false和"false") true/false是布尔值, cout的默认输出为 "1" / "0" ; "true"/"false"是字符串常量, cout的默认输出为字符串输出, 即为"true"/"false"</div> <div>3、进阶思考: 目前直接输出逻辑常量true和false, 在屏幕上输出的输出是1/0, 如果想输出为true/false, 应该怎么做?</div> <div>注意: 1、不允许用分支语句/条件运算符 2、提示: 去网上查一个前导格式控制符(课件无)</div> <div>加一个前导格式控制符boolalpha, 即 cout <<boolalpha<< true << endl;</div> <div>cout << boolalpha << false << endl;</div>
--	---



§. 基础知识题 - 关系运算、逻辑运算与选择结构

3、逻辑常量与逻辑变量

B. 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()
{
    bool k1 = true;

    cout << sizeof(true) << endl;
    cout << sizeof(k1) << endl;
    cout << k1 << ' ' << int(k1) << endl;

    cout << endl;

    bool k2 = false;
    cout << sizeof(false) << endl;
    cout << sizeof(k1) << endl;
    cout << k2 << ' ' << int(k2) << endl;

    return 0;
}
```

1、贴运行结果

2、bool型常量/变量在内存中占用_1_字节，值是_1/0_

总结bool型常量/变量在输出时的规则
(限制：在无3.A的前导格式控制符的前提下)
true的输出为“1”，false的输出为“0”



§. 基础知识题 - 关系运算、逻辑运算与选择结构

3、逻辑常量与逻辑变量

C. 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()
{
    bool k;

    cin >> k;
    cout << k << ' ' << int(k) << endl;

    return 0;
}
```

1、输入0，输出是：

2、输入1，输出是：

3、输入123，输出是：

4、输入true，输出是：

5、输入false，输出是：

总结bool型变量在输入时的规则：

bool变量在输入的时候会读取一个整型数据；非0的话则读取为true，0则读取为false。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

3、逻辑常量与逻辑变量

D. 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()
{
    bool k;

    k='A';
    cout << k << ' ' << (int)k << endl;

    k=0;
    cout << k << ' ' << (int)k << endl;

    k=256;
    cout << k << ' ' << (int)k << endl;

    char c = 256;
    cout << (int)c << endl;

    return 0;
}
```

1、贴运行结果



2、解释VS下warning的意思

在赋值的过程中, 发生了int/char到bool类型的转换, 会造成一定的信息截断损失

3、k='A' 是1字节赋值给1字节, 为什么还有warning?

因为bool类型的数据仅有1/0; 从char向bool类型的转换时也会发生信息的截断损失

4、k=256如果按整型的4字节赋给1字节, k应该是多少?

现在实际是多少? 为什么? (和c对比)

应该是0; 实际上是1, 因为bool仅有1/0两种值, 在赋值的过程中按照“非0为真0为假”的原则, 即原值为非0的赋值为1

5、为什么不 cout << c, 而是 (int)c ?

因为从char的cout默认输出为字符串型, 如果cout<<c会默认得到一个字符的输出而不是相对应的整型值

6、“非0为真0为假”这句话如何解释?

在其他的数据类型向bool类型转换的过程中, 原值二进制形式为0的转换为false, 非0的转换为true

§. 基础知识题 - 关系运算、逻辑运算与选择结构



3、逻辑常量与逻辑变量

E. 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()
{
    bool f=true;
    int a=10;

    a=a+f;
    cout << a << endl;

    return 0;
}
```

1、贴运行结果



2、当bool参与表达式计算时，当做 1/0来计算

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

A. 完成下列两个表格的填写（a/b是两个逻辑值，填写的内容不要用黑色）

a	b	!a	!b	a&&b	a b
1	1	0	0	1	1
1	0	0	1	0	1
0	1	1	0	0	1
0	0	1	1	0	0

a	b	!a	!b	a&&b	a b
非0	非0	0	0	1	1
非0	0	0	1	0	1
0	非0	1	0	0	1
0	0	1	1	0	0



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

B. 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)

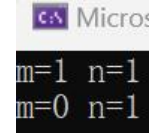
```
#include <iostream>
using namespace std;

int main()
{
    int a=1, b=2, c=3, d=4, m=1, n=1;

    cout << "m=" << m << " n=" << n << endl;
    (m=a>b)&&(n=c>d);
    cout << "m=" << m << " n=" << n << endl;

    return 0;
}
```

1、贴运行结果



m=1 n=1
m=0 n=1

2、解释 $(m=a>b)\&\&(n=c>d)$ 的求值过程 (标出步骤顺序)

(1) $a>b \Rightarrow \text{false}(0)$

(2) $m=a>b \Rightarrow \text{false}(0)$

(3) $(m=a>b) \Rightarrow \text{false}(0)$

(4) $(m=a>b)\&\&(\textcolor{red}{n=c>d}) \Rightarrow \text{false}(0)$

标红部分没有实际运算, 因为前面一个表达式的值已经为0了, 已经可以确定这个表达式的值为0了。

3、短路运算的意思是: 在优先运算的一侧结果已经可以确定整个表达式的结果时, 对于另一侧剩下的部分不再进行计算, 保留原值。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

C. 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()
{
    return 0;
}
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 100;
7      5 > 3 && 2 || (a = (8 < 4 - !0));
8      cout << a << endl;
9      return 0;
10 }
11
12 Microsoft Visual Studio 调试控制台
13 100
```

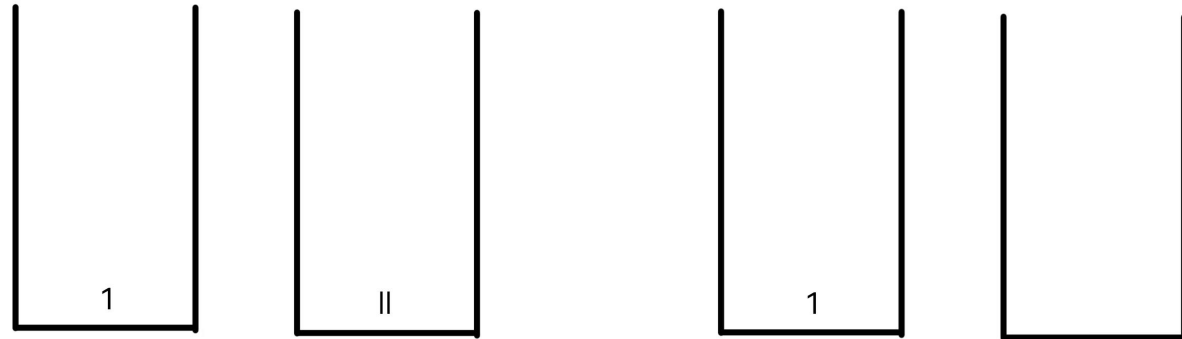
有以下逻辑表达式

$5 > 3 \ \&\& \ 2 \ || \ 8 < 4 - !0$

1、构造一个测试程序，在不改变该表达式目前求值顺序的情况下（允许插入新的运算，但目前这几个运算符的顺序不要变），证明 $8 < 4 - !0$ 存在短路运算

见左边的图

2、用栈方式画包含短路运算的表达式，则从分析到短路运算符进栈开始（本例中为 $||$ ），忽略_每个_运算符。（所有 / 比 $||$ 优先级高的）



§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

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()
{
    int i;

    cout<<"请输入成绩[0-100]"<<endl;
    cin >> i;

    if (i<60) {
        cout << "不及格" << endl;
    }
    cout << "程序结束" << endl;

    return 0;
}
```

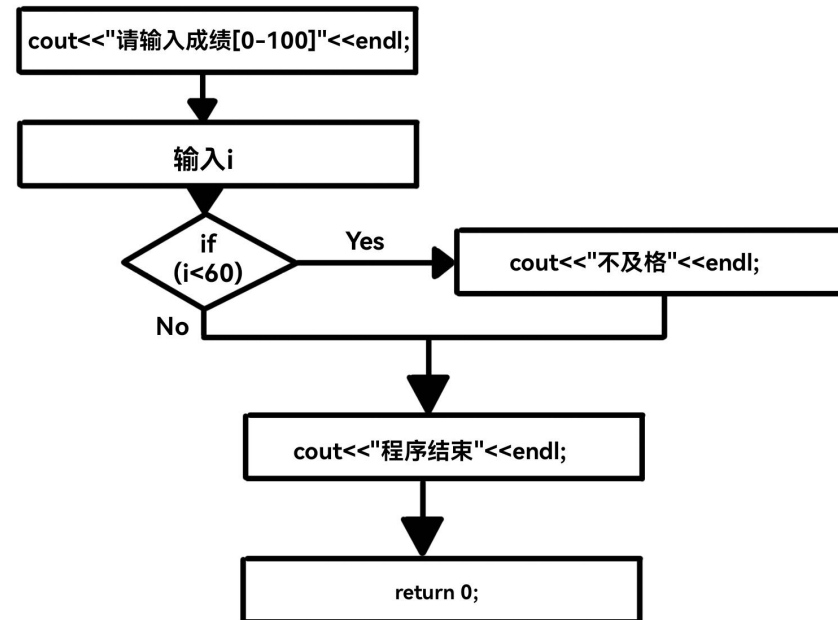
1、输入34，贴运行结果

```
Microsoft Visual Studio
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
Microsoft Visual Studio 调
请输入成绩[0-100]
74
程序结束
```

3、画出程序对应的流程框图





§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

B. 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()
{
    int i;

    cout<<"请输入成绩[0-100]"<<endl;
    cin >> i;

    if (i<60) {
        cout << "不及格" << endl;
        cout << "程序结束" << endl; //未缩进
    }

    return 0;
}
```

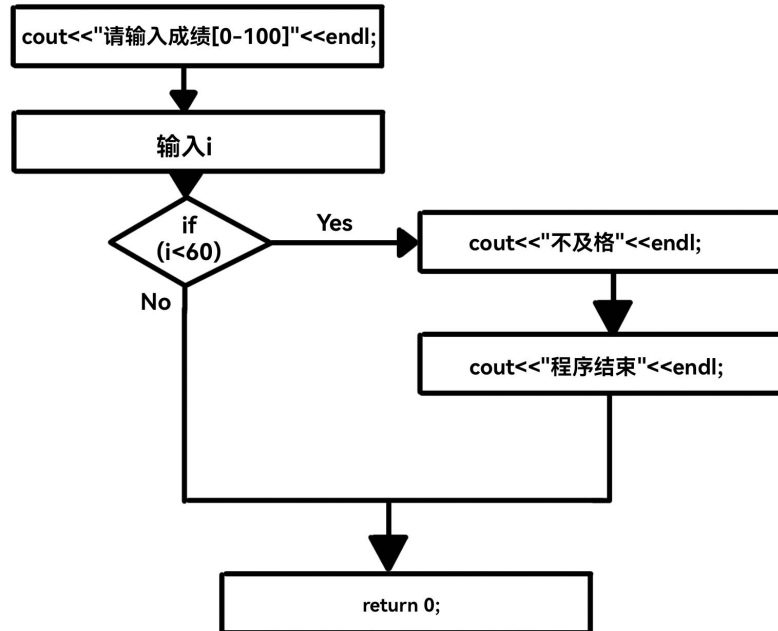
1、输入34，贴运行结果

```
Microsoft Visual Studio
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
Microsoft Visual Studio
请输入成绩[0-100]
74
```

3、画出程序对应的流程框图



4、程序标注“未缩进”的行，_应该_（应该/不应该）缩进



§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

C. 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()
{
    int i;

    cout<<"请输入成绩[0-100]"<<endl;
    cin >> i;

    if (i<60;) {
        cout << "不及格" << endl;
        cout << "程序结束" << endl; //未缩进
    }

    return 0;
}
```

贴编译错误并给出解释



if后面的()里应该是bool类型的表达式而不是语句, 左侧程序中在if后面的()里添加了“;”, 导致这一个表达式变成了语句, 不符合语法规则, 所以报错。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

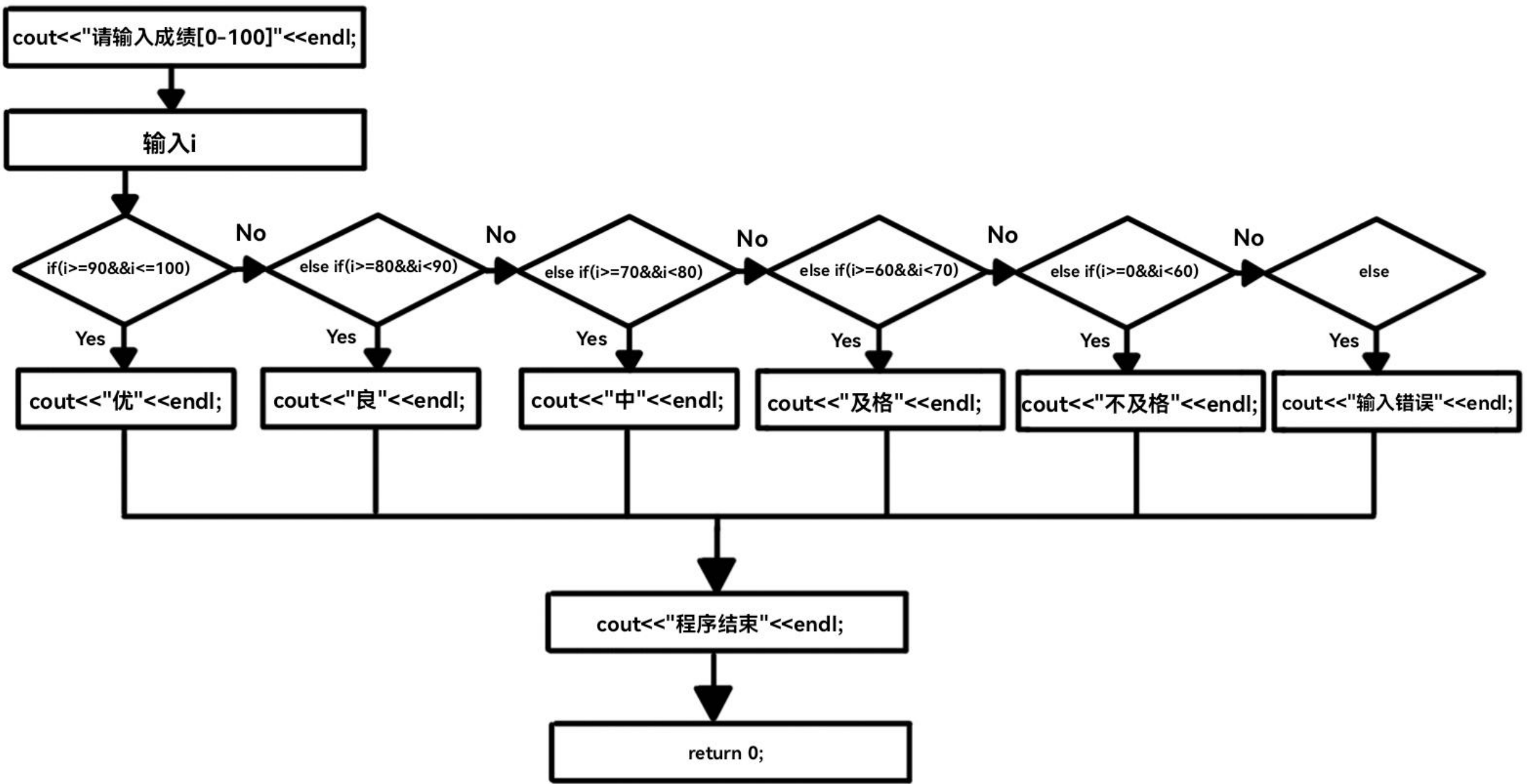
5、if语句 - 基本使用

D. 观察下列程序的运行结果，回答问题

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

int main()
{
    int i;
    cout << "请输入成绩[0-100]" << endl;
    cin >> i;
    if (i>=90 && i<=100)
        cout << "优" << endl;
    else if (i>=80 && i<90)
        cout << "良" << endl;
    else if (i>=70 && i<80)
        cout << "中" << endl;
    else if (i>=60 && i<70)
        cout << "及格" << endl;
    else if (i>=0 && i<60)
        cout << "不及格" << endl;
    else
        cout << "输入错误" << endl;
    cout << "程序结束" << endl;
    return 0;
}
```

- 1、给出程序的流程框图(注意字体的清晰可辨)
见下一页。
- 2、 $i < 90$ 能否改为 $i \leq 89$? 哪个更好?
在本道题目中可以改;
 $i < 90$ 更好, 首先和上下的格式更一致, 输入更加的简洁;
而且如果把分数换成double型此程序依旧可以用, 即能运用更加广泛。
- 3、 $i < 90$ 能否改为 $i \leq 90$? 运行是否正确?
能, 正确。



§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套

A. 一个有10行代码的if语句嵌套，回答问题

0: if (表达式) { 1: if (表达式) { 2: } 3: else { 4: } 5: } 6: else { 7: if (表达式) { 8: } 9: }	第0行的"{" 和 第_5_行的"}"配对 第1行的"{" 和 第_2_行的"}"配对 第3行的"{" 和 第_4_行的"}"配对 第6行的"{" 和 第_9_行的"}"配对 第7行的"{" 和 第_8_行的"}"配对 总结：给出大括号配对的基本准则 自上而下, 按照类似于栈的方式两两配对
---	---

§. 基础知识题 - 关系运算、逻辑运算与选择结构



6、if语句 - 多重嵌套

B. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    B;  
}
```

1、当表达式1__真__(真/假/任意)，表达式2_真__(真/假/任意)时，
执行语句A

2、当表达式1__真__(真/假/任意)，表达式2_任意__(真/假/任意)时，
执行语句B



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套
C. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    else {  
        B;  
    }  
    C;  
}  
else {  
    if (表达式3) {  
        D;  
    }  
    E;  
}
```

- 1、当表达式1__真____(真/假/任意)，表达式2__真____(真/假/任意)时，
 执行语句A
- 2、当表达式1__真____(真/假/任意)，表达式2__假____(真/假/任意)时，
 执行语句B
- 3、当表达式1__假____(真/假/任意)，表达式2__任意____(真/假/任意)时，
 执行语句C
- 4、当表达式1__假____(真/假/任意)，表达式3__真____(真/假/任意)时，
 执行语句D
- 5、当表达式1__假____(真/假/任意)，表达式3__任意____(真/假/任意)时，
 执行语句E



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套

D. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    else {  
        B;  
    }  
    C;  
}  
→ F;  
else {  
    if (表达式3) {  
        D;  
    }  
    E;  
}
```

在6.C的基础上，在箭头位置插入语句F

1、请构造一个符合此要求的测试程序，并给出该程序的程序及编译错误截图

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main()  
5  {  
6      int x;  
7      cout << "给x赋值: " << endl;  
8      if (x>=0) {  
9          if (x>=1) {  
10             cout << "A" << endl;  
11         }  
12         else {  
13             cout << "B" << endl;  
14         }  
15         cout << "C" << endl;  
16     }  
17     cout << "D" << endl;  
18     else {  
19         if (x<=-1) {  
20             cout << "E" << endl;  
21         }  
22         cout << "F" << endl;  
23     }  
24     return 0;  
25 }  
26
```

2、请说明错误原因

if块和与其相对应的else块之间不能有其他语句，必须连续，否则非法。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套
E. 一个if语句嵌套如下，回答问题

<pre>if (表达式1) { if (表达式2) { A; } B; } else { C; }</pre>	<p>左侧代码按缩进格式排版</p> <pre>if (表达式1) { if (表达式2) { A; } B; } else { C; }</pre>	<p>1、当表达式1_真_(真/假/任意)，表达式2_真_(真/假/任意)时，执行语句A</p> <p>2、当表达式1_真_(真/假/任意)，表达式2_任意_(真/假/任意)时，执行语句B</p> <p>3、当表达式1_假_(真/假/任意)，表达式2_任意_(真/假/任意)时，执行语句C</p>
<pre>if (表达式1) { if (表达式2) { A; } else { B; } C; }</pre>	<p>左侧代码按缩进格式排版</p> <pre>if (表达式1) { if (表达式2) { A; } else { B; } C; }</pre>	<p>1、当表达式1_真_(真/假/任意)，表达式2_真_(真/假/任意)时，执行语句A</p> <p>2、当表达式1_真_(真/假/任意)，表达式2_假_(真/假/任意)时，执行语句B</p> <p>3、当表达式1_真_(真/假/任意)，表达式2_任意_(真/假/任意)时，执行语句C</p>

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

7、条件运算符与条件表达式

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()
{
    int a,b;

    cin >> a >> b;

    if (a>b)
        cout << "max=" << a << endl;
    else
        cout << "max=" << b << endl;

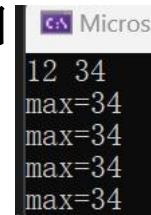
    a > b ? cout << "max=" << a << endl : cout << "max=" << b << endl; //1

    cout << "max=" << (a>b?a:b) << endl; //2

    printf("max=%d", a>b?a:b); //3

    return 0;
}
```

1、输入12 34，给出运行截图



```
12 34
max=34
max=34
max=34
max=34
max=34
```

2、输入34 12，给出运行截图



```
34 12
max=34
max=34
max=34
max=34
max=34
```

3、//1 //2 //3这三种条件运算符的使用，按你的喜欢程度排序为_2>3>1_



§. 基础知识题 - 关系运算、逻辑运算与选择结构

7、条件运算符与条件表达式

B. 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()
{
    int a=1, b=2;

    a==1 ? "Hello" : 123;

    a>b ? cout << a : printf("%d", b);

    a==1 ? 'A' : 123;

    return 0;
}
```

1、给出编译报错的截图



2、条件表达式使用的三句中，前两句报错，最后一句正确，总结下条件表达式使用时的限制规则（提示：注意表达式2和表达式3的类型）

条件表达式中的形式：
表达式1? 表达式2: 表达式3;
其中表达式1、2、3的类型可以不同，但是表达式2、3的类型必须1相同。

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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)

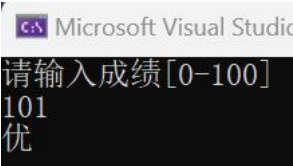
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
    }

    return 0;
}
```

程序的期望，是当输入的score在[0..100]时，分段输出“优/良/中/及格/不及格”，否则输出“输入错误”

1、程序不完全正确，找出不符合期望的两个数据区间并给出运行截图
(不需要改对)

[101, 109]
[-9, -1]





§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

B. 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()
{
    const int k=5;
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
        case k+2:
            cout<<"中"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
    }

    return 0;
}
```

在8.A的基础上
1、将6、8、default的位置进行了交换
2、将7写为常变量+常量形式

验证此程序与8.A的功能是否完全一致
(即：8.A中正确的, 此程序中同样正确；8.A错误的, 此程序中同样错误)



结论：8.A和8.B__完全一致__(完全一致/不完全一致)
如果不完全一致，给出表现不一致的测试数据的运行截图



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

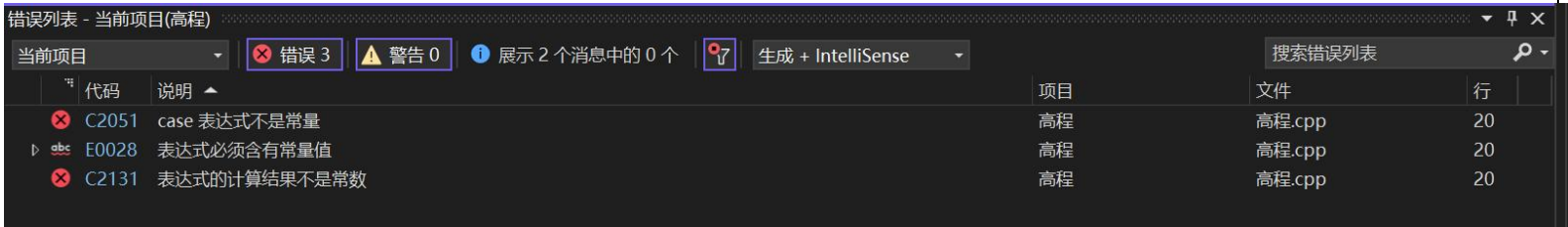
C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
int main()
{
    int k=5;
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
        case k+2:
            cout<<"中"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
    }

    return 0;
}
```

在8.B的基础上，将k从const int改为int

1、给出编译错误的截图



2、解释错误原因

case后面必须是整型(表达式)常量, k是变量, 表示整型(表达式)常量, 所以会报错。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

D. 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()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
        case 4+2:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
    }

    return 0;
}
```

在8.A的基础上，多了一个case 4+2

1、给出编译错误的截图

代码	说明	项目	文件	行
C2196	case 值"6"已使用	高程	高程.cpp	20
E1578	case 标签值已经出现在此开关 所在行数:19 中	高程	高程.cpp	20

2、解释错误原因

多个case后面所跟的常量表达式的值必须互不相同,但是左侧的程序中case后面出现的6和4+2是两个相同的常量表达式,所以报错。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

E. 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()
{
    float score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
    }

    return 0;
}
```

在8.A的基础上，将score从int改为float

1、给出编译错误的截图

当前项目 错误 13 警告 0 展示 1 个消息中的 0 个 生成 + IntelliSense

		搜索错误列表		
代码	说明	项目	文件	行
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	13
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	16
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	19
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	22
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	23
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	24
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	25
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	26
E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	高程	高程.cpp	27
C2450	类型为 "float" 的 switch 表达式无效	高程	高程.cpp	8
E0847	表达式必须包含整数或枚举类型	高程	高程.cpp	8

2、解释错误原因

case后面必须是整型常量表达式, 但是左侧程序中case后面跟的是浮点表达式, 所以报错。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

F. 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()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;

        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
    }

    return 0;
}
```

在8.A的基础上，删除case 8后面的break

1、给出与8.A运行结果不一致的测试数据即截图



2、解释break的作用
跳出所在那一层的switch-case表达式



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

G. 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()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout<<"输入错误"<<endl;
            break;
    }

    return 0;
}
```

程序同8. A，将其改正确，即所有[0..100]之外的数据均给出“输入错误”即可

```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout << "请输入成绩[0-100]" << endl;
    cin >> score;
    if (score > 100 || score < 0)
    {
        cout << "输入错误" << endl;
        return 0;
    }
    switch (score / 10) {
        case 10:
        case 9:
            cout << "优" << endl;
            break;
        case 8:
            cout << "良" << endl;
            break;
        case 7:
            cout << "中" << endl;
            break;
        case 6:
            cout << "及格" << endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout << "不及格" << endl;
            break;
        default:
            cout << "输入错误" << endl;
            break;
    }

    return 0;
}
```




§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

H. 思考

如果将成绩区间对应为:

[85-100]	- 优
[70-85)	- 良
[60-70)	- 及格
[0-60)	- 不及格

1、用if-else语句完成该程序并贴图

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int score;
6     cout<<"请输入成绩[0 - 100]"<<endl;
7     cin >> score;
8     if (score > 100 || score < 0)
9         cout << "输入错误" << endl;
10    if (score >= 85 && score <= 100)
11        cout << "优" << endl;
12    if (score >= 70 && score < 85)
13        cout << "良" << endl;
14    if (score >= 60 && score < 70)
15        cout << "及格" << endl;
16    if (score >= 0 && score < 60)
17        cout << "不及格" << endl;
18    return 0;
19 }
20
```

2、如果用switch语句, 该如何实现? (如果程序太长, 允许只截取能说明问题的部分即可)

3、如果学生成绩带小数点, 即"xx.5"形式, 能用if语句吗? 能用switch语句吗? 请解释原因
if可以, switch不可以; switch只能对于整型数据的判断跳转。

4、总结switch语句使用时的注意事项

switch只能对于整型数据的判断跳转; switch只能适用于特定的一个值 (并不是一个特定的范围) 进行跳转; switch后面记得要加break; switch后面的case语句标签不能重复。

5、switch-case语句能完全取代if-else吗?

不能

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int score;
6     cout << "请输入成绩[0-100]" << endl;
7     cin >> score;
8     if (score > 100 || score < 0)
9     {
10         cout << "输入错误" << endl;
11         return 0;
12     }
13     switch (score / 5)
14     {
15         case 20:
16         case 19:
17         case 18:
18         case 17:
19             cout << "优" << endl;
20             break;
21         case 16:
22         case 15:
23         case 14:
24             cout << "良" << endl;
25             break;
26         case 13:
27         case 12:
28             cout << "及格" << endl;
29             break;
30         default:
31             cout << "不及格" << endl;
32             break;
33     }
34     return 0;
35 }
```

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目