

§. 基础知识题 – 关系运算、逻辑运算与选择结构



要求:

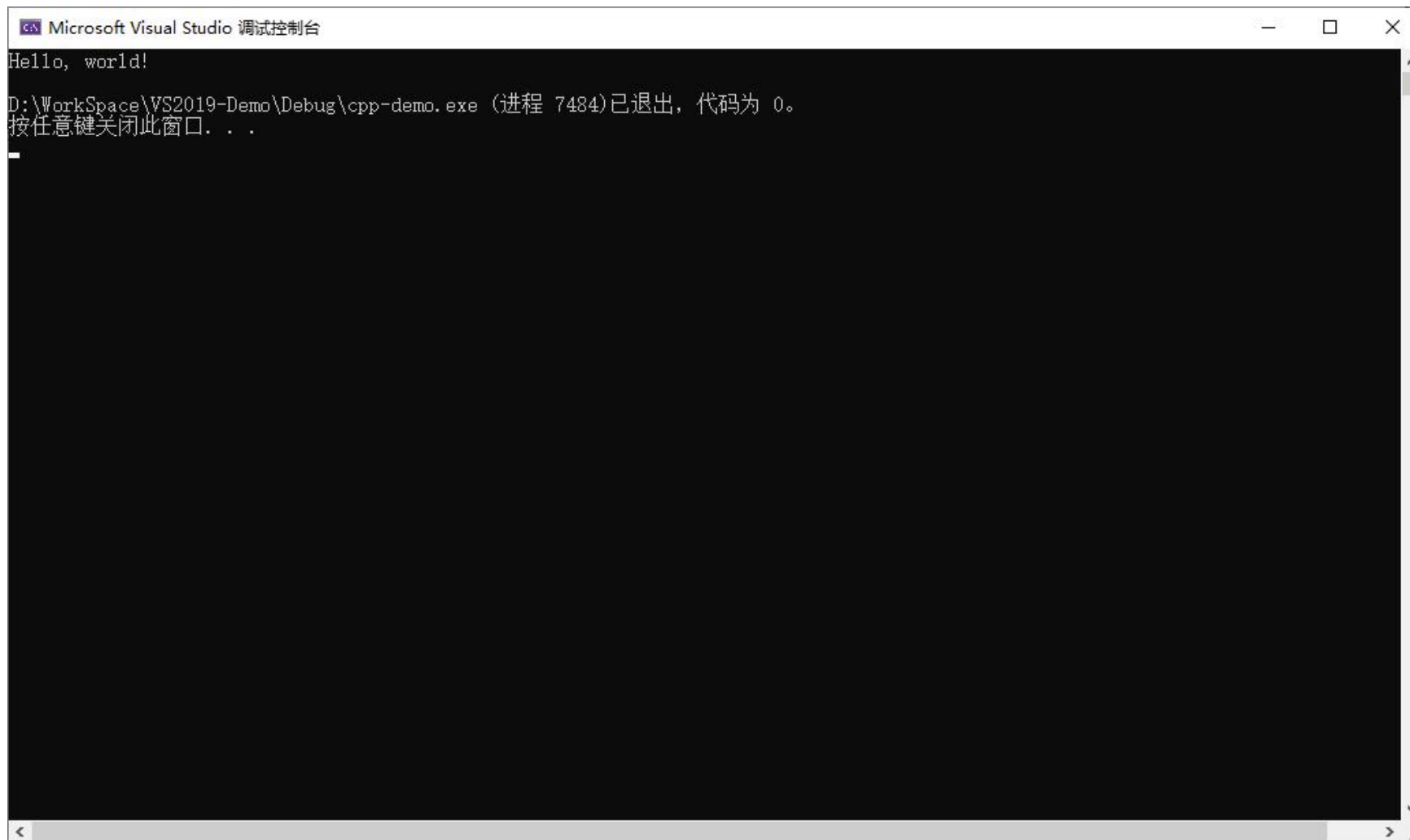
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2019编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**10月7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

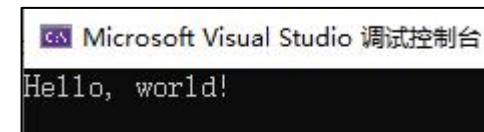


贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

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 text "Hello, world!" followed by "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 context of the output.

例：有效贴图

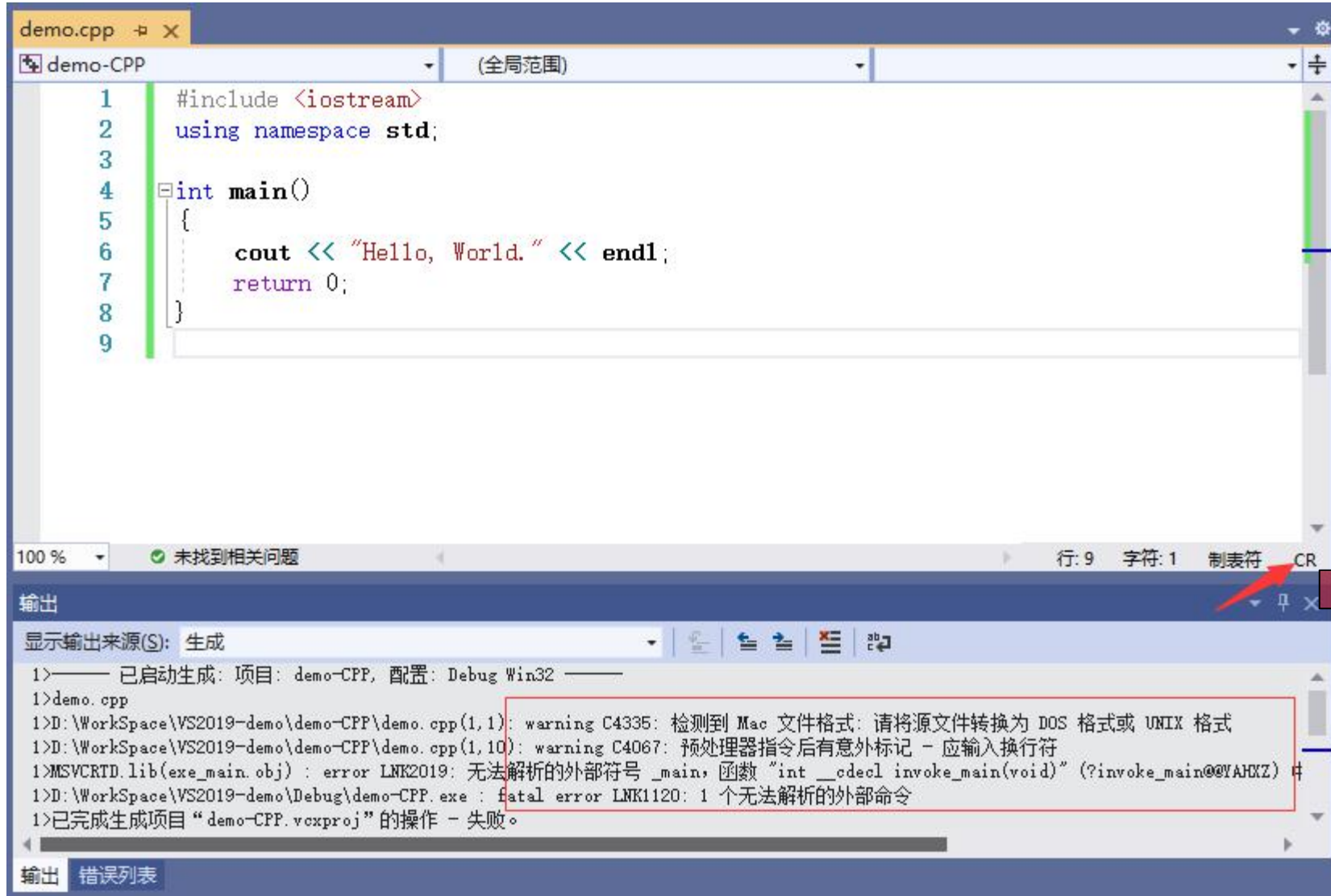
A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the "Hello, world!" text. The window title "Microsoft Visual Studio 调试控制台" is visible at the top.



§. 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VS2019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

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;

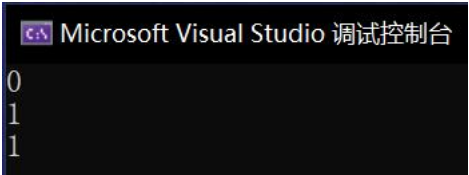
    d = a > b > c;
    cout << d << endl;

    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

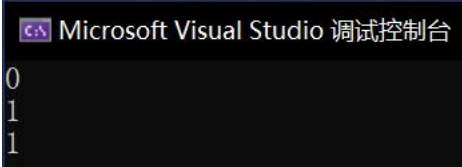
答：整个判断过程是先判断左边的关系运算，若正确，关系运算表达式的值转换为1，再判断右边的关系运算；若错误，关系运算表达式的值转换为0，再判断右边的关系运算。而不是按我们的思路，在a、b、c三个变量中去比较，所以会对结果产生影响。



§. 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)

<pre>#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; }</pre>	<div>1、贴运行结果</div> <div></div> <div>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？ $a < b < c$这个式子，按常规理解，$3 < 2 < 1$是错误的，为什么结果是1？ $b > a < c$这个式子，按常规理解，$2 > 3 < 1$是错误的，为什么结果是1？</div> <div>答：整个判断过程是先判断左边的关系运算，若正确，关系运算表达式的值转换为1，再判断右边的关系运算；若错误，关系运算表达式的值转换为0，再判断右边的关系运算。</div> <div>①$a > b > c$这个式子，$a > b$正确，转换为1，$1 > c$错误，故结果为0 ②$a < b < c$这个式子，$a < b$错误，转换为0，$0 < c$正确，故结果为1 ③$b > a < c$这个式子，$b > a$错误，转换为0，$0 < c$正确，故结果为1</div>
---	---

§. 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>
using namespace std;

int main()
{
    int a=1;
    cout << (a==1) << endl;

    float b=1.1f;
    cout << (b==1.1) << endl;

    double c=1.1;
    cout << (c==1.1) << endl;

    return 0;
}
```

贴运行结果

Microsoft Visual Studio 调试控制台

```
1
0
1
```



§. 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>
using namespace std;

int main()
{
    float b=1.1f;
    cout << (b==1.1) << endl;
    cout << (fabs(b-1.1)<1e-6) << endl;

    float c=1.0f;
    cout << (c==1.0) << endl;
    cout << (fabs(c-1.0)<1e-6) << endl;

    return 0;
}
```

1、贴VS+Dev下的运行结果

Microsoft Visual Studio 调试控制台

```
0
1
1
1
1
```

D:\Safe Zone\homework\test\test1.exe

```
0
1
1
1
1
```

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

Microsoft Visual Studio 调试控制台

```
0
1
1
1
1
```

信息

In function 'int main()':

[Error] 'fabs' was not declared in this scope; did you mean 'labs'?

3、综合2. A和2. B，实数进行相等比较时的通用方法是比较两数在精度范围内是否相等



§. 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

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>
#include <cmath> //VS2019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;
} //VS2019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？

答：float相同，double不同。

因为double型浮点数有效位数是15位，故小数点后第6位、第7位都被赋值到d1、d2上了

```
d1=123.456789012346
d2=123.456789123457
fabs(d1-d2)=0.000000111111
(fabs(d1-d2)<1e-6)=1
(fabs(d1-d2)<1e-7)=0
```

而float型有效位数是6位，赋值时产生截断，故f1、f2均只有六位有效数字

```
f1=123.457
f2=123.457
fabs(f1-f2)=0.000
(fabs(f1-f2)<1e-6)=1
(fabs(f1-f2)<1e-7)=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

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)

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

int main()
{
    cout << true    << endl;
    cout << "true"  << endl;

    cout << endl;

    cout << false   << endl;
    cout << "false" << endl;

    return 0;
}
```

```
test1.cpp x
test (全局范围) main()
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << true << endl;
7     cout << boolalpha << true << endl;
8     cout << false << endl;
9     cout << noboolalpha << false << endl;
10    cout << true << endl;
11
12    return 0;
13 }
```

1、贴运行结果

```
Microsoft Visual Studio 调试控制台
1
true
0
false
```

2、解释 true 和 "true" 的区别 (false和"false")

答: true为bool型常量, 数值为1; "true"为字符串常量, 输出为一串字符; false为bool型常量, 数值为0; "false"为字符串常量, 输出为一串字符。

3、进阶思考: 如果想使true和false在屏幕上输出true/false, 应该怎么做?

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

答: 使用前导格式控制符boolalpha, 它对之后输出的所有数据都生效, 除非数据类型不是bool型, 或使用另一前导格式控制符noboolalpha解除(它也对之后输出的所有数据都生效)。



§. 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;

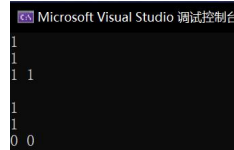
    cout << sizeof(true) << endl;
    cout << sizeof(k1) << endl;
    cout << k1 << ' ' << int(k1) << endl;

    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或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)

答：常量：true输出为1，false输出为0
变量：判断变量的值是true还是false，若为true，输出为1；若为false，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)

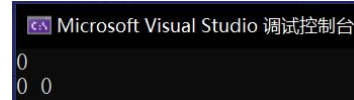
```
#include <iostream>
using namespace std;

int main()
{
    bool k;

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

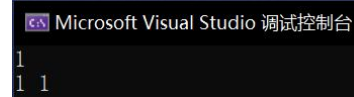
    return 0;
}
```

1、输入0，输出是：



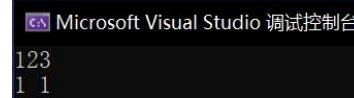
```
Microsoft Visual Studio 调试控制台
0
0 0
```

2、输入1，输出是：



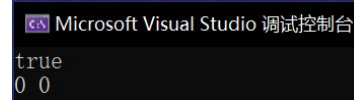
```
Microsoft Visual Studio 调试控制台
1
1 1
```

3、输入123，输出是：



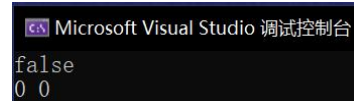
```
Microsoft Visual Studio 调试控制台
123
1 1
```

4、输入true，输出是：



```
Microsoft Visual Studio 调试控制台
true
0 0
```

5、输入false，输出是：



```
Microsoft Visual Studio 调试控制台
false
0 0
```

总结bool型变量在输入时的规则：

答：1、输入数时，非0为真0为假

2、无法通过直接输入true/false为bool型变量赋值



§. 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=123;
    cout << k << ' ' << int(k) << endl;

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

    return 0;
}
```

1、贴运行结果

```
Microsoft Visual Studio 调试控制台
1 1
0 0
```

2、解释VS下warning的意思

答：“123”为int型整数，大小为4字节，而k为bool型变量，大小仅为1字节，赋值的时候可能会截断，产生错误。

3、“非0为真0为假”这句话如何解释？

答：给bool型变量直接按照数字的形式赋值时，若该数为0，bool型变量的值为假；若该数不为0，bool型变量的值为真。



§. 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

Microsoft Visual Studio 调试控制台
11

2、当bool参与表达式计算时，当做数值0或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

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）

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

```
Microsoft Visual Studio 调试控制台
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)

答：① $m=a$ 真， $(m=a)=1$
② $1>b$ 假， $(m=a>b)=0$
③ $0\&\&(n=c>d)$ 假， $(m=a>b)\&\&(n=c>d)=0$

3、短路运算的意思是：只有当必须执行下一个逻辑运算符才能求出解时，才执行该运算符，否则不执行。
如本例中求得 $(m=a>b)=0$ 后， $(m=a>b)\&\&(n=c>d)$ 的值一定为 0，故程序不再求解 $(n=c>d)$ 的值

§. 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;
}
```

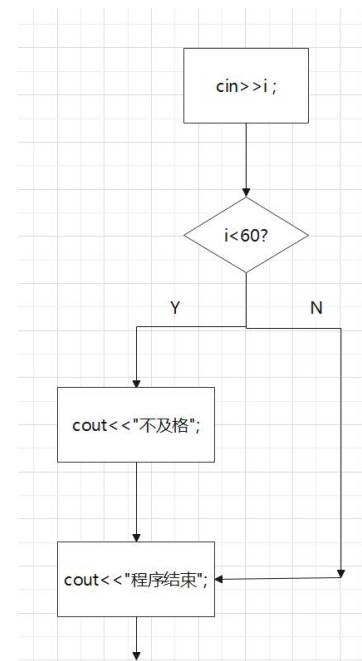
1、输入34，贴运行结果

```
Microsoft Visual Studio 调试控制台
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 调试控制台
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;
}
```

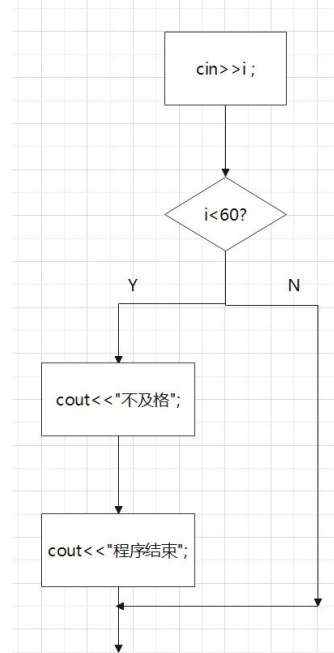
1、输入34，贴运行结果

```
Microsoft Visual Studio 调试控制台
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 调试控制台
请输入成绩[0-100]
74
程序结束
```

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

代码	说明
E0018	应输入")"
C4552	"<": 未使用表达式结果
C2429	语言功能 "if/switch 中的 init-statement" 需要编译器标志 "/std:c++17"
C2059	语法错误: ")"
C2143	语法错误: 缺少";"(在"{"的前面)

解释：if后的括号里应该跟运算表达式，而不是语句，加；产生了语法错误。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

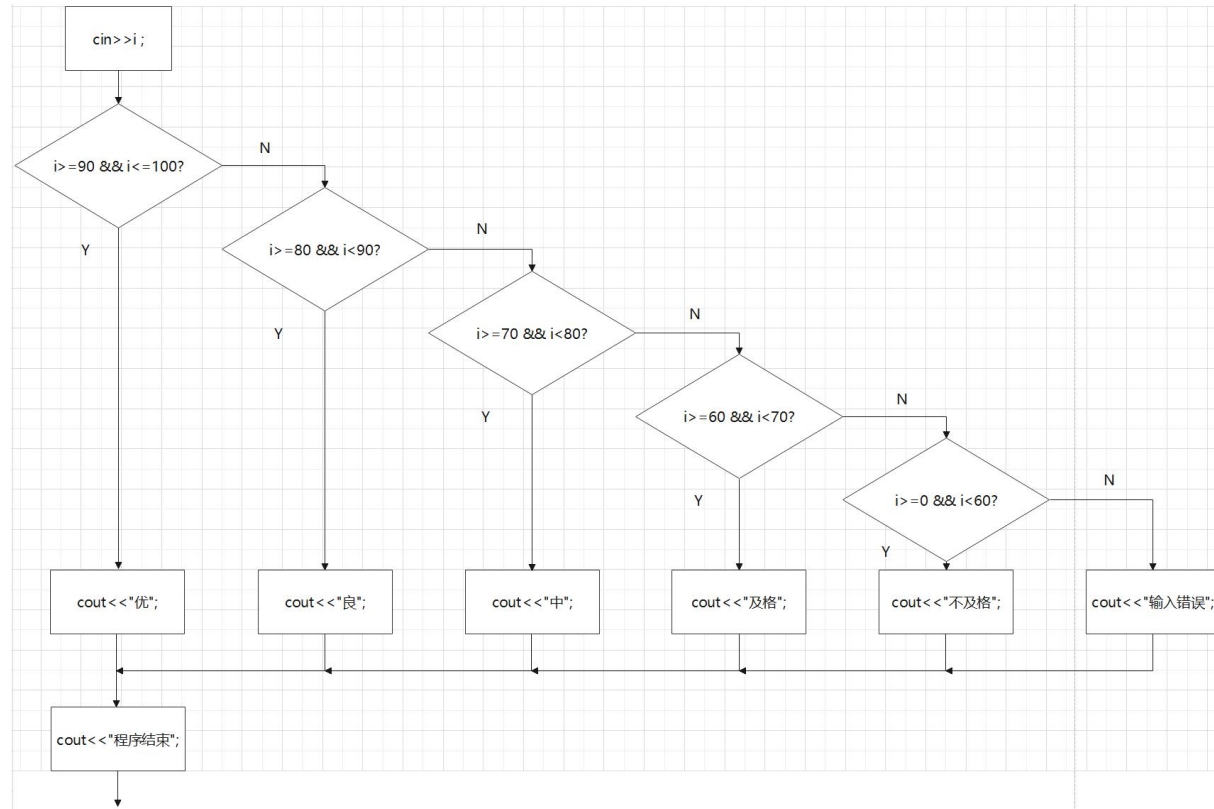
5、if语句 - 基本使用

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>=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)



2、i<90能否改为i<=89? 哪个更好?

答: 如果只改动这一处可以。但i<90更好, 因为如果后续若把i改为浮点数, 如果改动, 输入区间(89, 90)的数, 得到的输出结果就不符合预期了。

3、i<90能否改为i<=90? 运行是否正确?

答: 能, 运行正确。因为输入90时符合第一个if语句的判断条件, 输出了“优”, 不再进行下一个else if语句的判断, 不必担心因条件改变而造成输入90输出“良”的情况。

§. 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 答：自上而下，忽略“{”，以“}”为准向上与第一个未匹配的“}”匹配
---	--

§. 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真(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，并给出该程序的编译错误截图

2、请说明错误原因

答：整个双分支算一个语句，中间不能插入其他语句。插入了其他语句F，编译器会认定前面的if语句已结束，else没有与其匹配的if，故出现错误。

The screenshot shows a C++ IDE with a file named test1.cpp. The code is as follows:

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main()  
5 {  
6     int i;  
7     cout << "请输入成绩[0-100]" << endl;  
8     cin >> i;  
9     if (i >= 80 && i <= 100) {  
10  
11         if (i >= 90 && i <= 100) {  
12             cout << "优" << endl;  
13         }  
14  
15         else {  
16             cout << "良" << endl;  
17         }  
18  
19         cout << "恭喜您，您对知识掌握得很好！" << endl;  
20     }  
21  
22     cout << "您未进入最佳分数段，请稍等....." << endl;  
23  
24     else {  
25  
26         if (i >= 70 && i <= 80) {  
27             cout << "中" << endl;  
28         }  
29  
30         cout << "及格或不及格，您要加油啦！" << endl;  
31     }  
32 }
```

The IDE shows two compilation errors in the error list at the bottom:

- 错误 2: E0127 应输入一个语句 (at line 24, column 12)
- 错误 2: C2181 没有匹配 if 的非法 else (at line 24, column 12)



§. 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

```
Microsoft Visual Studio 调试控制台
12 34
max=34
max=34
max=34
max=34
```

2、输入34 12，给出运行截图

```
Microsoft Visual Studio 调试控制台
34 12
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、//1、//3



§. 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;

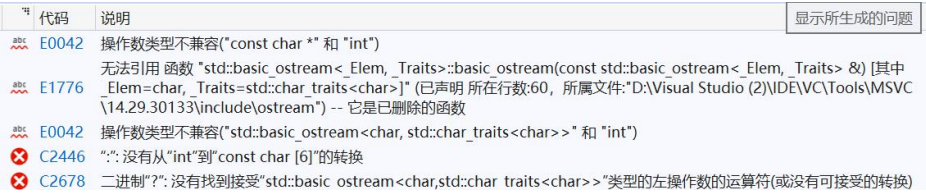
    a==1 ? "Hello" : 123; //编译报错

    a>b ? cout << a : printf("%d", b); //编译报错

    a==1 ? 'A' : 123; //编译正确

    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)

答：使用条件表达式中的三目运算符时，“:”连接的两个表达式类型必须一致。

§. 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

区间：(100, 110)和(-10, 0)





§. 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： 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

代码	说明
E0028	表达式必须含有常量值
C2131	表达式的计算结果不是常数
C2051	case 表达式不是常量

2、解释错误原因

答：case后应接整型常量或整型常量+整形常变量，但此时k的类型是int，是整型变量，不符合语句使用要求，产生了语法错误。



§. 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
E1578	case 标签值已经出现在此开关 所在行数:19 中
C2196	case 值"6"已使用

2、解释错误原因

答：每个case后的整型常量表达式的值应各不相同，而4+2=6，所以case 4+2与case 6重复，产生了错误。



§. 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

代码	说明
E0847	表达式必须包含整数或枚举类型
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
E2373	此常量表达式的类型为 "int", 而所需类型为 "float"
C2450	类型为 "float" 的 switch 表达式无效

2、解释错误原因



答: switch后括号内应为整型常量表达式, 但此时score/10为float型/int型, 结果应该是float型, 不符合语句使用要求, 产生了错误。此外, score/10无法与int型相等, 无法达到任何一个case的条件(两者类型不相同), 产生了错误。



§. 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)

<pre>#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; }</pre>	<p>在8.A的基础上，删除case 8后面的break</p> <p>1、给出与8.A运行结果不一致的测试数据即截图</p> <div><div>8. A</div></div> <div><div>8. F</div></div> <p>2、解释break的作用</p> <p>答：分隔各个case语句，使各语句在不同条件下分别执行，而不是连续执行。</p>
---	--



§. 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)

<pre>#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; }</pre>	<p>程序同8. A，将其改正确，即符合预期的期望</p> <pre>#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: score > 100 ? cout << "输入错误" << endl : cout << "优" << endl; break; 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: cout << "不及格" << endl; break; case 0: score >= 0 ? cout << "不及格" << endl : cout << "输入错误" << endl; break; default: cout << "输入错误" << endl; break; } return 0; }</pre>
---	---





§. 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

```
test1.cpp  x
test (全局范围)
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int i;
7      cout << "请输入成绩[0-100]" << endl;
8      cin >> i;
9      if (i >= 85 && i <= 100)
10         cout << "优" << endl;
11     else if (i >= 70 && i < 85)
12         cout << "良" << endl;
13     else if (i >= 60 && i < 70)
14         cout << "及格" << endl;
15     else if (i >= 0 && i < 60)
16         cout << "不及格" << endl;
17     else
18         cout << "输入错误" << endl;
19     cout << "程序结束" << endl;
20     return 0;
21 }
```

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）

```
switch (score / 5) {
case 20:
    score > 100 ? cout << "输入错误" << endl : cout << "优" << endl;
    break;
case 19:
case 18:
case 17:
    cout << "优" << endl;
    break;
case 16:
case 15:
case 14:
    cout << "良" << endl;
    break;
case 13:
case 12:
    cout << "及格" << endl;
    break;
```

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

答：可以用if语句，因为if语句是在判断范围的基础上执行语句，完全可以应付浮点型数据；但是不可用switch语句，因为switch语句的判断条件是整型变量，学生成绩若不改变类型，无法达到判断标准

4、总结switch语句使用时的注意事项

答：1、判断条件一定是整型常量、整形常量或两者相加，否则会报error
2、若想让各语句在不同条件下分别执行，一定要记得加break语句
3、每个case后的整型常量表达式的值应各不相同
4、若对输入的值做处理（比如扩大n倍）后再进行判断，switch-case语句可能会扩大范围，使一些区间的数的输出结果不符合预期，故一定要结合问题实际，时刻注意语句应用的范围。

5、switch-case语句能完全取代if-else吗？

答：不能完全替代。if-else语句在一些方面比起switch-case语句还是有优势的，比如本例中if-else语句可以直接判断一个数是否在一个范围内，但如果想让switch-case语句达到同样的效果，就需要对数据进行处理，使多个数据的处理结果等于同一个整型结果，从而达到要求，否则工作量就会很大；此外，switch-case语句使用时的注意事项也不少，一不小心就会使运行结果不符合预期，初学时使用起来可能不比if-else语句方便。所以在我看来，switch-case无法完全取代if-else。

§. 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