

§. 基础知识题 - 关系运算、逻辑运算与选择结构



要求:

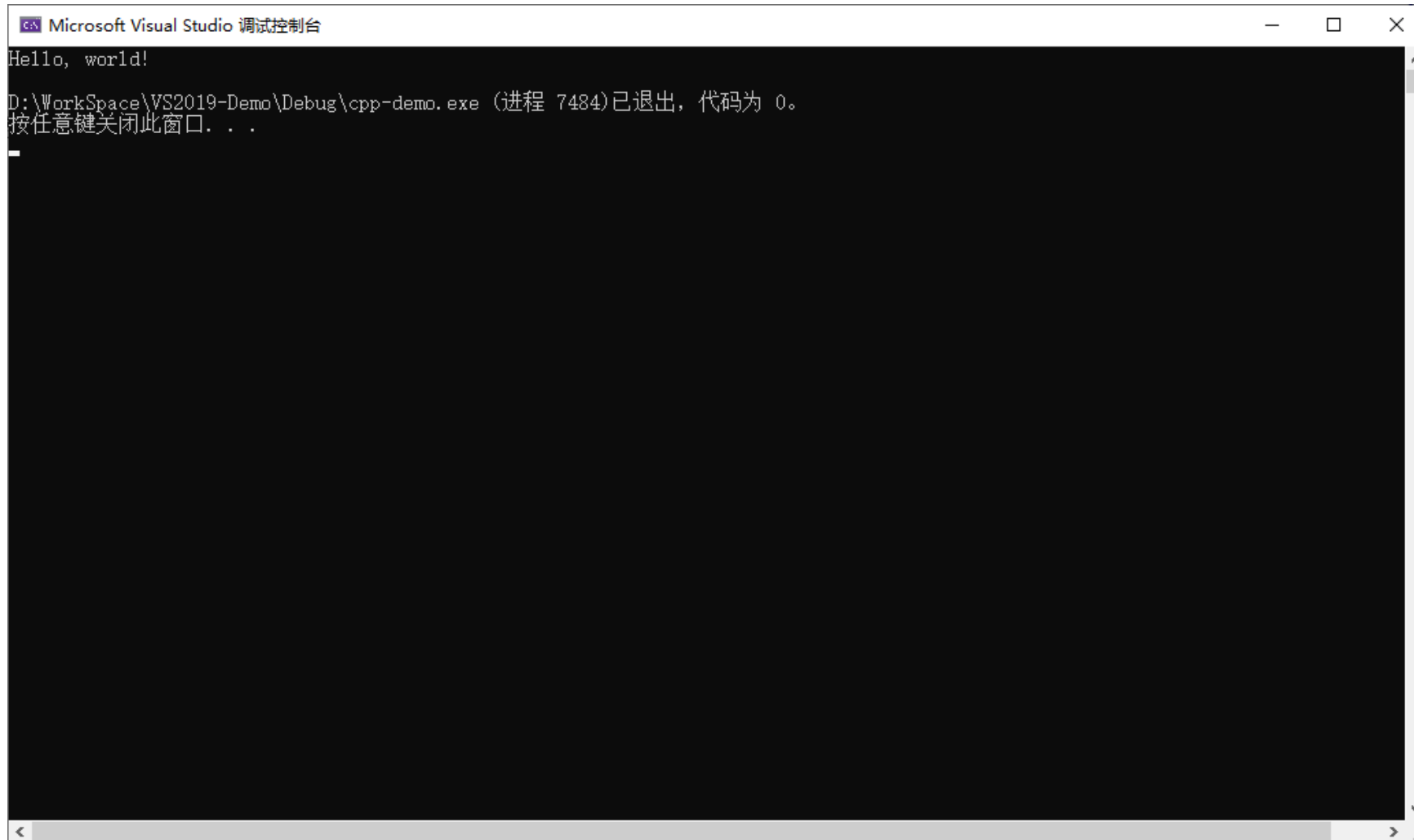
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**9月29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

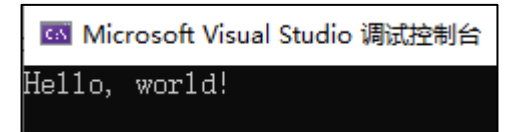
贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图



```
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



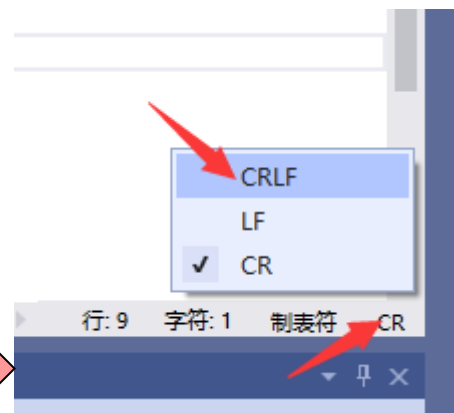
```
Microsoft Visual Studio 调试控制台
Hello, world!
```



§. 基础知识题 - 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

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



以d=a>b>c为例，该式子相当于d=((a>b)>c)
(a>b)是一个关系表达式，结果为0，是一个bool值，将bool值与int值比较大小会引发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;

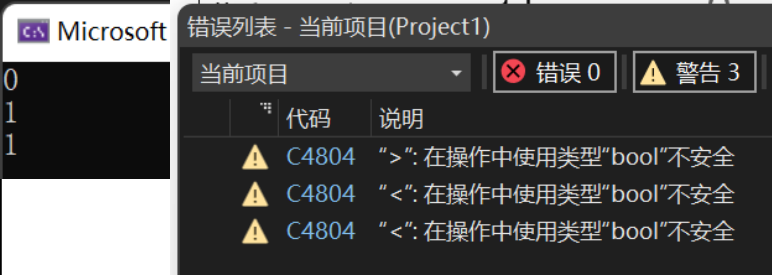
    d = a > b > c;
    cout << d << endl;

    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？
- $a < b < c$ 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？

- (1) 在程序中， $a > b > c$ 相当于 $(a > b) > c$ ，而 $(a > b)$ 是一个关系表达式，结果为1， $1 > c$ 不成立，故最终结果为0；
- (2) 在程序中， $a < b < c$ 相当于 $(a < b) < c$ ，而 $(a < b)$ 是一个关系表达式，结果为0， $0 < c$ 成立，故最终结果为1；
- (3) 在程序中， $b > a < c$ 相当于 $(b > a) < c$ ，而 $(b > a)$ 是一个关系表达式，结果为0， $0 < c$ 成立，故最终结果为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 b = 1.1;
    cout << (b - 1.1) << endl;
    cout << (b == 1.1) << endl;
    cout << (fabs(b - 1.1) < 1e-6) << endl;

    float c = 1.0;
    cout << (c - 1.0) << endl;
    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

Vs:

Dev:

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

Vs:

Dev: 程序编译错误，无法运行。

3、由本例得出的结论，实数进行相等比较时的通用方法是 _若两个实数的差的绝对值小于某个很小的数（如1e-6），那么就认为这两个实数相等_



§. 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

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 $\text{fabs}(**)<1e-6$ 和 $\text{fabs}(**)<1e-7$ 在float和double下的表现，哪个相同？哪个不同？为什么？

$\text{fabs}(**)<1e-6$ 表现相同， $\text{fabs}(**)<1e-7$ 表现不同。
因为double的有效位数为15~16位，float的有效位数为6~7位，在这个例子中，float的精度达不到 $1e-7$ ，所以对 $\text{fabs}(f1-f2)<1e-7$ 的判断不一定可信。



§. 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;

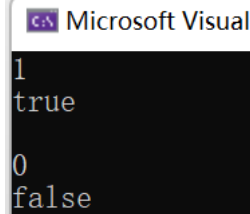
int main()
{
    cout << true    << endl;
    cout << "true"  << endl;

    cout << endl;

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

    return 0;
}
```

1、贴运行结果



```
1
true
0
false
```

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

true是一个bool值，相当于1； "true" 是一个字符串。

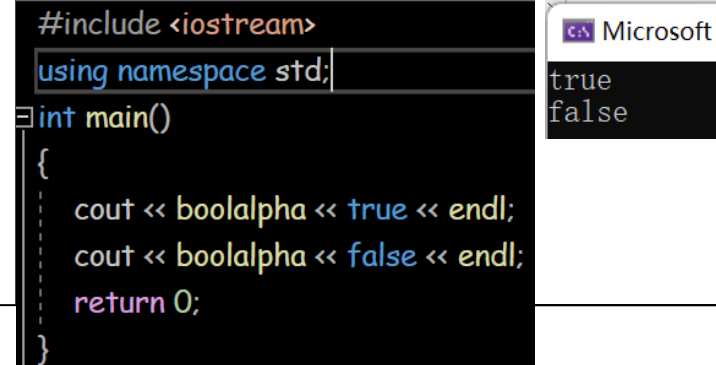
false是一个bool值，相当于0； "false" 是一个字符串。

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

利用前导格式控制符boolalpha。



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

int main()
{
    cout << boolalpha << true << endl;
    cout << boolalpha << false << endl;
    return 0;
}
```

```
true
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

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

```
Microsoft Visual
1
1
1 1
1
1
0 0
```

2、bool型常量/变量在内存中占用_1_字节，值是_0或1_

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)

若直接输出bool类型的量，都会输出0或1（true对应1，false对应0），而不会输出true或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

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：

1. 输入为数字，若数字非0，则bool值赋为1；
若数字为0，则bool值赋为0。
2. 输入非数字 bool值赋为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

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;

    return 0;
}
```

1、贴运行结果

```
Microsoft
warning C4305: "=" : 从 "char" 到 "bool" 截断
warning C4305: "=" : 从 "int" 到 "bool" 截断
1 1
0 0
1 1
```

2、解释VS下warning的意思

将char型或int型数据赋给bool型时，由于char型、int型的范围比bool型大，所以数据可能会部分丢失，需要warning。

3、k='A' 是1字节赋值给1字节，为什么还有warning?

bool型虽然是1字节，但是它的值只能是0或1，比char类型的范围小，故将1字节的char型赋给bool型时有warning。

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?

应该为0，现在实际为1。

因为给bool赋值的规律不是复制二进制表示，而是：非零值赋1，零值赋0。

5、“非0为真0为假”这句话如何解释?

如果给bool型赋非0数字，则bool值为1；否则为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

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)

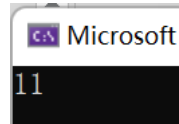
```
#include <iostream>
using namespace std;

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__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，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)

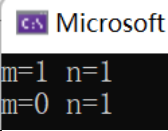
```
#include <iostream>
using namespace std;

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



2、解释 (m=a>b)&&(n=c>d) 的求值过程 (标出步骤顺序)

- ① a>b 结果为0
- ② m=a>b 将 (a>b) 的结果赋给m, 结果为0, m变为0
- ③ (m=a>b) && (n=c>d) 虽然还没有计算n=c>d的值, 但是由于已经得到(m=a>b)的值为0, 所以该表达式的结果必为0. 得到答案为0, 运算结束。

3、短路运算的意思是: __ 当一个关系表达式的值确定时, 运算结束 (无论是否将整个式子计算出来)。以&&为例, 若出现出现为0的值, 则&&的值必为0, 而无需继续全部计算完成。 __



§. 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()
{
    int x = 1e9;
    cout << (5 > 3 && 2 || (x = 8 < 4 - !0)) << endl;
    cout << x << endl;
    return 0;
}
```

Microsoft Visual Studio

```
1
1000000000
```

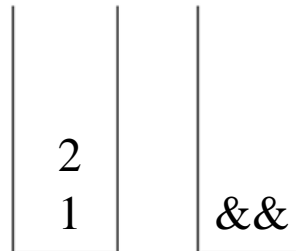
有以下逻辑表达式

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

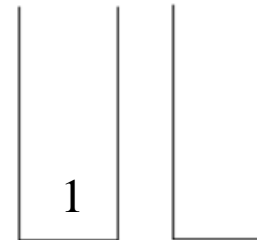
1、构造一个测试程序，在不改变该表达式目前求值顺序的情况下（允许插入新的运算，但目前这几个运算符的顺序不要变），证明 $8 < 4 - !0$ 存在/不存在短路运算

如左图，由于x的值没有改变，所以 $8 < 4 - !0$ 存在短路运算。

2、用栈方式画出从 $||$ 进栈开始（即第一张图为 $||$ 在栈顶），到整个表达式求值完成的过程（允许加页）



要进栈的 $(||)$ 低于栈顶的 $(\&\&)$ ，先计算



$(||)$ 满足进栈条件，但数字栈顶为1，直接得到答案1，运算结束



§. 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;
}
```

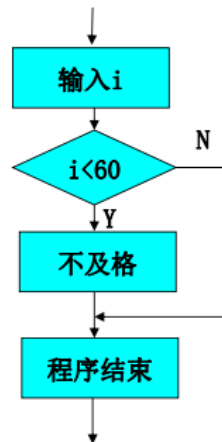
1、输入34，贴运行结果

```
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
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;
}
```

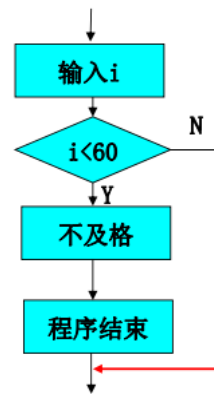
1、输入34，贴运行结果

```
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
C:\Users\DTTTTTT\Desktop
```

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;

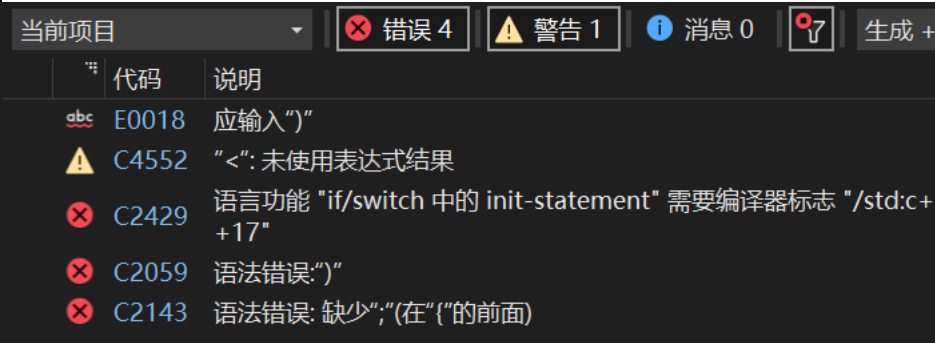
int main()
{
    int i;

    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



解释：if () 括号内的表达式后不能加分号；
表达式加了分号之后成为一个语句，而语句没有值，无法判断真假，故编译错误。

§. 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
```

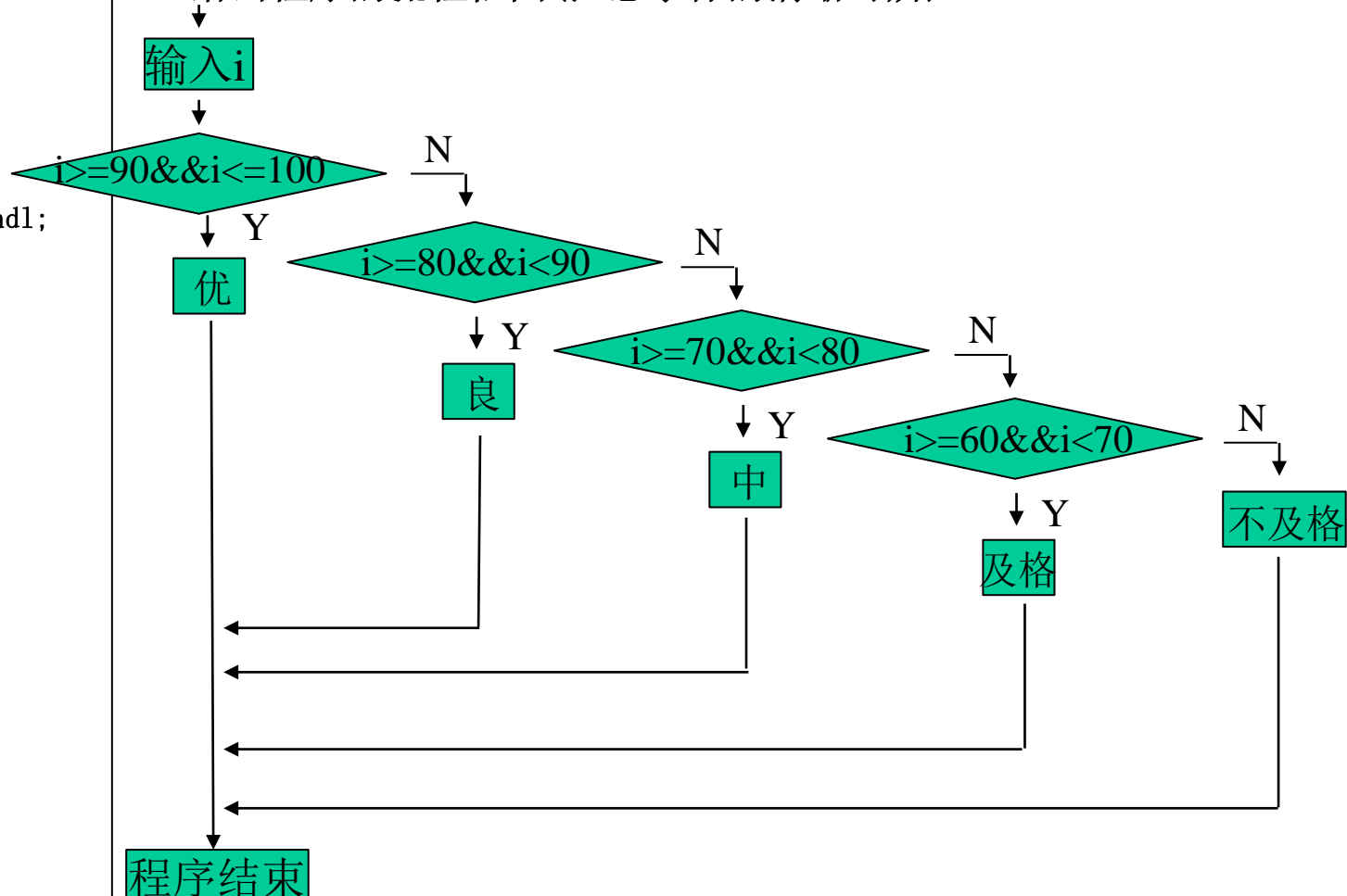
```
        cout << "输入错误" << endl;
```

```
    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)



§. 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;
}
```

2、 $i < 90$ 能否改为 $i \leq 89$? 哪个更好?

在此程序中改为 $i \leq 89$ 不影响程序正确性。

$i < 90$ 更好，因为在一些情况下分数可能为小数，这时 $i < 90$ 的表达式仍然适用

3、 $i < 90$ 能否改为 $i \leq 90$? 运行是否正确?

能，程序运行正确。

因为进入if ($i \geq 80$ && $i < 90$)语句的前提是($i \geq 90$ && $i \leq 100$)不成立，所以 i 不可能为90，故 $i < 90$ 与 $i \leq 90$ 在此处无差别。



§. 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

<pre>0: if (表达式) { 1: if (表达式) { 2: } 3: else { 4: } 5: } 6: else { 7: if (表达式) { 8: } 9: }</pre>	<p>第0行的"{" 和 第_5_行的"}"配对</p> <p>第1行的"{" 和 第_2_行的"}"配对</p> <p>第3行的"{" 和 第_4_行的"}"配对</p> <p>第6行的"{" 和 第_9_行的"}"配对</p> <p>第7行的"{" 和 第_8_行的"}"配对</p> <p>总结：给出大括号配对的基本准则</p> <p>自上而下，忽略 { ， 以 } 为准向上匹配第一个未配对的 { 。</p> <p>{ } 的匹配可用栈理解，遇 { 进栈，遇 } 则栈顶 { 出栈并匹配为一对, 若到最后仍有 { 或 } 未配对则语法错</p>
---	--



§. 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__随意__(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__随意__(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

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

```
int main()  
{  
    int x = 1, y = 1, z = 1, t = 0;  
    if (x == 1) {  
        if (y == 1) {  
            t = 1;  
        }  
        else {  
            t = 2;  
        }  
        t = 3;  
    }  
    t = 4;  
    else {  
        if (z == 1) {  
            t = 5;  
        }  
        t = 6;  
    }  
    return 0;  
}
```

abc E0127 应输入一个语句
✗ C2181 没有匹配 if 的非法 else

2、请说明错误原因

最后一个else前插入了一句t=4;
导致该else没有与之匹配的if.
故编译错误。



§. 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
12 34
max=34
max=34
max=34
max=34
```

2、输入34 12，给出运行截图

```
Microsoft Visual Studio
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 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;

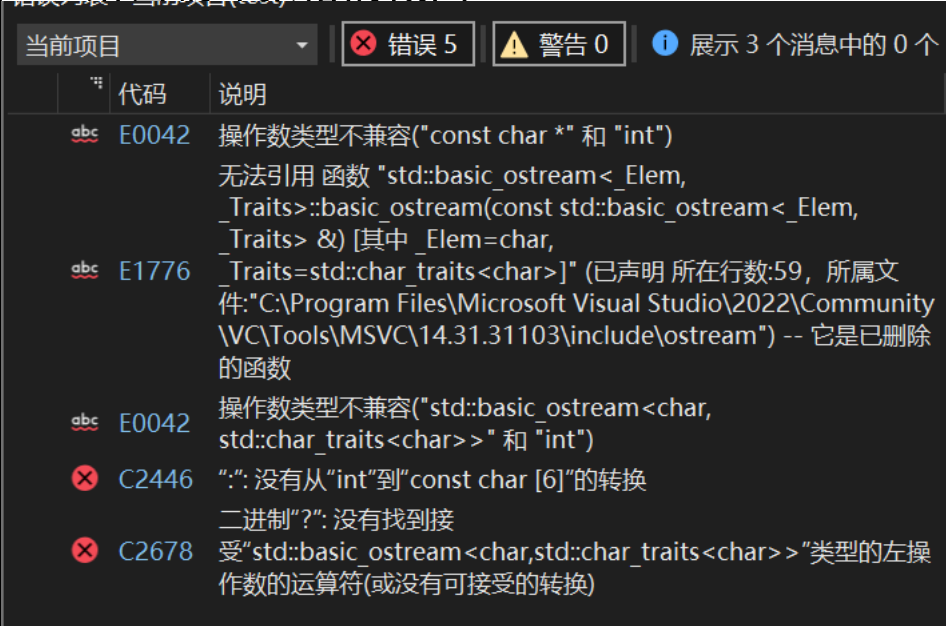
    a==1 ? "Hello" : 123; //编译报错

    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)

对于三目运算符 (表达式1) ? (表达式2) : (表达式3)
表达式1、2、3的类型可以不同，但2、3的类型必须相容
(否则无法确定条件表达式的值类型)



§. 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

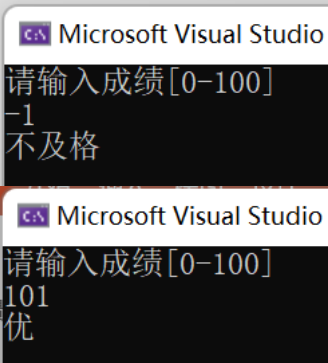
A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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
不符合期望的两个数据区间：(-10, 0) (100, 110)





§. 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)

<pre>#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; }</pre>	<p>在8. A的基础上</p> <ol style="list-style-type: none">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 <p>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)</p> <p>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</p>
--	---



§ . 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)

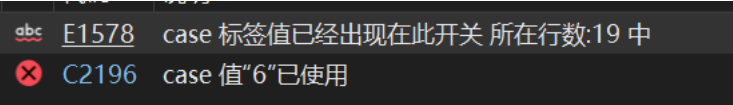
<pre>#include <iostream> using namespace std; int main() { int k=5; int score; cout<<"请输入成绩[0-100]"<<endl; cin >> score; switch(score/10) { case 10: case 9: cout<<"优"<<endl; break; case 6: cout<<"及格"<<endl; break; default: cout<<"输入错误"<<endl; break; case k+2: cout<<"中"<<endl; break; case 8: cout<<"良"<<endl; break; case 5: case 4: case 3: case 2: case 1: case 0: cout<<"不及格"<<endl; break; } return 0; }</pre>	<p>在8.B的基础上，将k从const int改为int</p> <p>1、给出编译错误的截图</p> <div></div> <p>2、解释错误原因</p> <p>k是int型变量，故（k+2）的值不是常量，而case表达式必须是常量，故编译错误。</p>
--	--



§. 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)

<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: 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; }</pre>	<p>在8. A的基础上，多了一个case 4+2</p> <p>1、给出编译错误的截图</p>  <p>2、解释错误原因</p> <p>4+2=6 相当于整个switch中有两个case 6 而case后的表达式不能出现两个相同的值，否则会导致二义性 故编译错误。</p>
---	---



§. 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



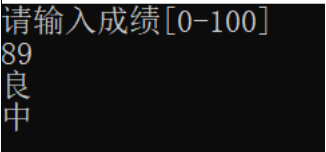
2、解释错误原因
switch表达式不能是浮点类型，浮点类型的switch表达式无效，故编译错误。



§. 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>  <p>2、解释break的作用 跳出switch语句。 若没有break, 则会一直往下执行语句直到switch语句结束或遇到下一个break。</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; if (score >= 0 && score <= 100) { 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; } } else { cout << "输入错误" << endl; } 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

```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout << "请输入成绩[0-100]" << endl;
    cin >> score;
    if (score < 0 || score > 100)
        cout << "输入有误";
    else {
        if (score >= 85)
            cout << "优" << endl;
        else if (score >= 70)
            cout << "良" << endl;
        else if (score >= 60)
            cout << "及格" << endl;
        else
            cout << "不及格" << endl;
    }
    return 0;
}
```

2、如果用switch语句，该如何实现？

```
if (score < 0 || score > 100)
    cout << "输入有误";
else {
    switch (score * 2 / 10) {
        case 20:
        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;
        default:
            cout << "不及格" << endl;
            break;
    }
}
```



§. 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

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

能用if语句，不能用switch语句。因为if表达式只要能判断真假即可，数据类型可以任意选择，但是switch表达式不能是浮点类型。

4、总结switch语句使用时的注意事项

- ① 表达式不可以是浮点类型
- ② case后必须是一个常量
- ③ case后的常量不能有重复
- ④ 一个case后的语句执行完后要写break;

5、switch-case语句能完全取代if-else吗？
不能。

一方面，并不是所有的判断条件都能方便地转化成整数类型的值；

另一方面， switch-case语句要枚举出所有的可能值，在很多情况下过程较繁琐。



§. 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