

§. 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

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 context of the program's execution and termination.

例：有效贴图

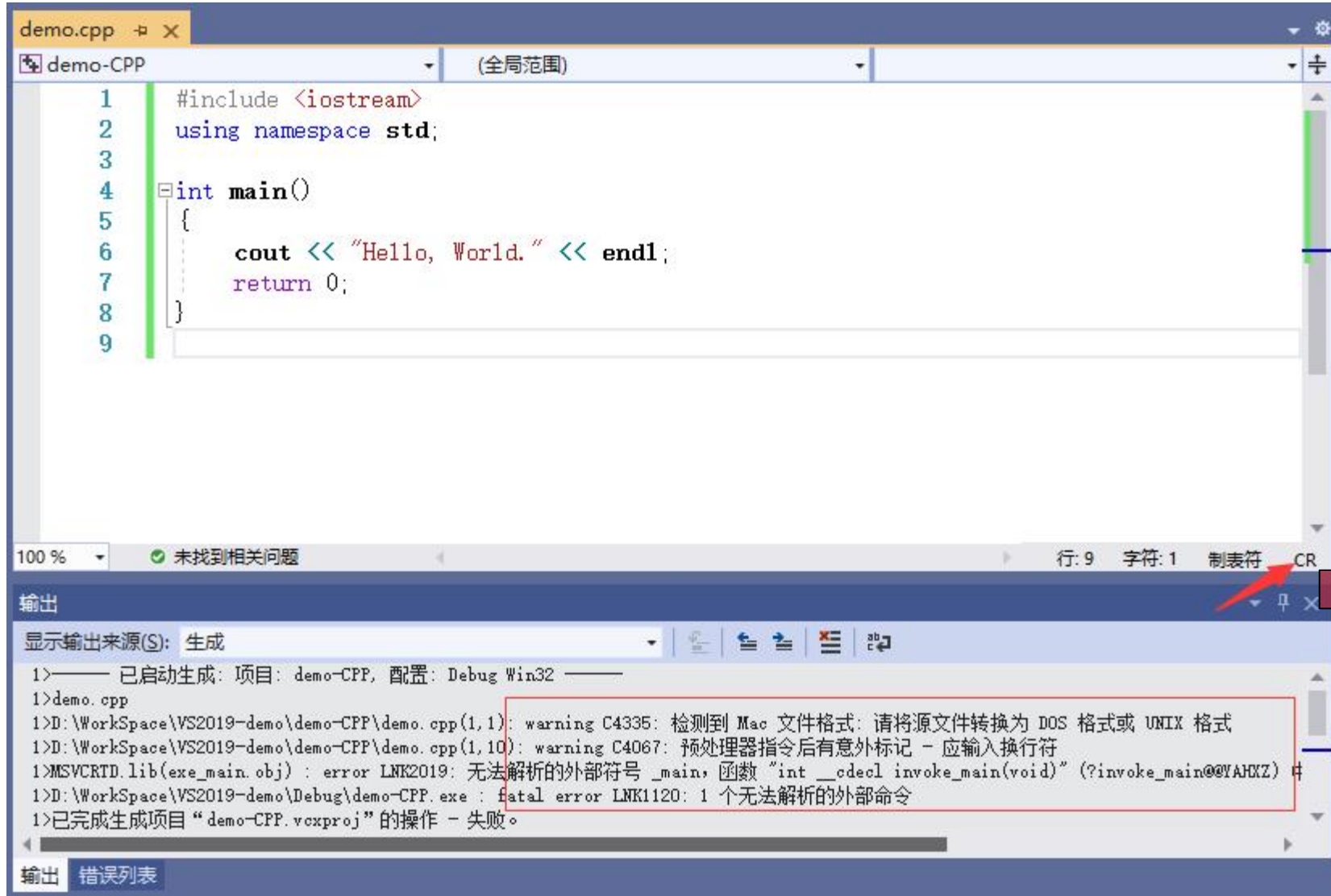
A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the output text. The window is titled "Microsoft Visual Studio 调试控制台". It contains the text: "Hello, world!". This is an example of a valid screenshot as it focuses on the output without unnecessary window details.



§. 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;

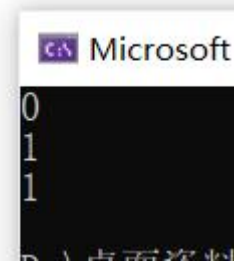
    d = b > a < c;
    cout << d << endl;

    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

因为我们的<和>是关系运算符，结果是0或1，系统认为我们用他们并不能表示a, b, c三者的大小关系，并不安全。





§. 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？
- 因为这个式子是右结合，先计算 $3 > 2$ ，结果是0或1，然后再和1去比。
- 因为这个式子是右结合，先计算 $3 < 2$ ，结果是0或1，然后再和1去比。
- 因为这个式子是右结合，先计算 $2 > 3$ ，结果是0或1，然后再和1去比。



§. 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;

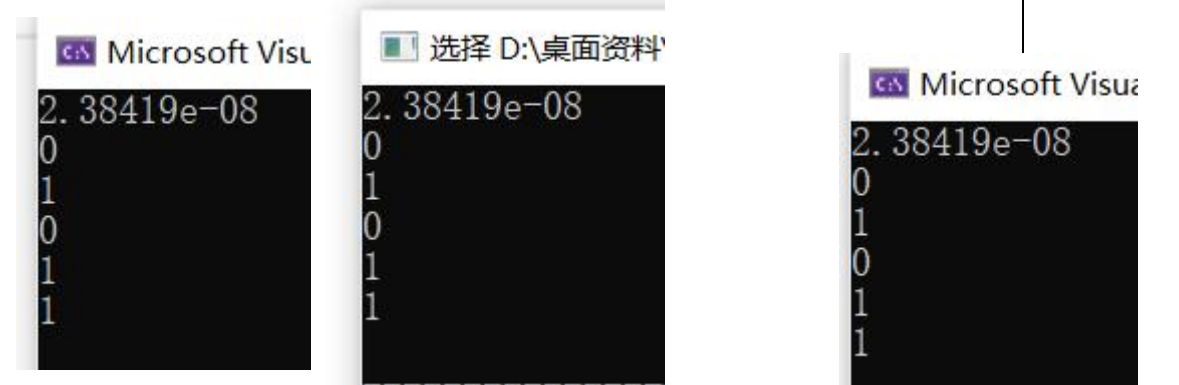
    float c = 1.0;
    cout << (c - 1.0) << endl;
    cout << (c == 1.0) << endl;
    cout << (fabs(c - 1.0) < 1e-6) << endl;

    return 0;
}
```

1、贴VS+Dev下的运行结果

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____采用浮点数绝对值的方法____





§. 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;

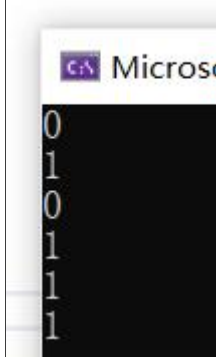
int main()
{
    double d1=123.456789012345678;
    double d2=123.456789123456789;
    cout << (d1==d2) << endl;
    cout << (fabs(d1-d2)<1e-6) << endl;
    cout << (fabs(d1-d2)<1e-7) << endl;

    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？



1e-7的时候是不同的，而1e-6的时候是相同的。

因为float型的数据没有double型数据的精度高，所以在使用绝对值精度时会发生不同。

§. 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

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)

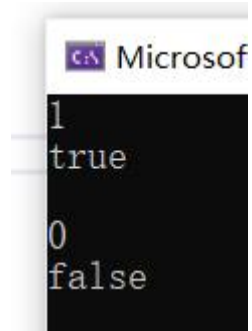
```
#include <iostream>
using namespace std;

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

    cout << endl;

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

    return 0;
}
```



1、贴运行结果

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

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

2. 第一个true是逻辑常量，所以输出的时候是1
而第二个true是字符串的形式输出的，会整个一起输出。

3. 可以使用 `cout<<boolalpha<<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;

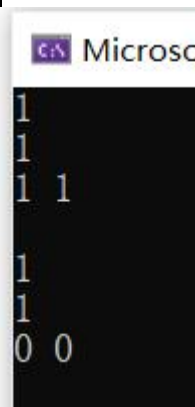
    return 0;
}
```

1、贴运行结果

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)

根据它的值输出0或者1





§. 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)

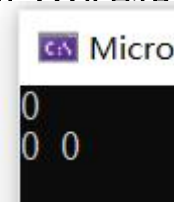
```
#include <iostream>
using namespace std;

int main()
{
    bool k;

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

    return 0;
}
```

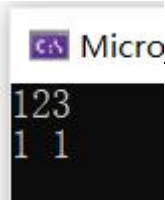
1、输入0，输出是：



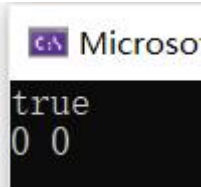
2、输入1，输出是：



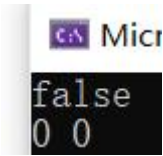
3、输入123，输出是：



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：

在输入数字时会按照首位进行读取，输入true和false时按照0来处理。



§. 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;

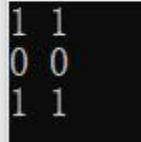
    k='A';
    cout << k << ' ' << (int)k << endl;

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

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

    return 0;
}
```

1、贴运行结果



2、解释VS下warning的意思

在给变量赋初值的时候，发生了变量类型的转换，数据类型发生了截断。

3、k='A' 是1字节赋值给1字节，为什么还有warning?

当字符量赋给bool型的数据时，会根据是否非0，决定赋值，相对的对char型数据是一种截断。

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0，但实际上是1

5、“非0为真0为假”这句话如何解释?

在bool型的赋值运算中，不是按照之前的多字节赋值给少字节的方法，发生截断，二是根据非0为真0为假的规则。



§. 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来（如果右错则贴错误信息截图）

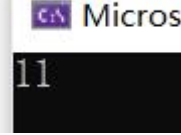
```
#include <iostream>
using namespace std;

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__int型的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

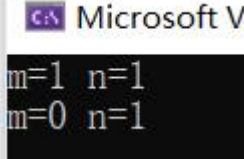
a	b	!a	!b	a&&b	a b
非0	非0	0	0	0	0
非0	0	0	1	0	1
0	非0	1	0	0	1
0	0	1	1	1	0



§. 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)

<pre>#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; }</pre>	<div>1、贴运行结果</div> <div></div> <div>2、解释 (m=a>b)&&(n=c>d) 的求值过程（标出步骤顺序） 先计算a>b，然后计算赋值语句m=0。 根据逻辑运算符&&，后面的表达式不再计算。</div> <div>3、短路运算的意思是：<u>当&&和 运算符的结果已经确定时，就不会计算后面的表达式，相当于被短路了</u></div>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



§. 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;
}
```

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

int main()
{
    int m = 15;
    5 > 3 && 2 || (m = (8 < 4 - !0));
    cout << m << endl;
    return 0;
}
```

Microsoft Visual Studio 调试控制

15

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/不存在短路运算发生短路运算

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）



§. 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;

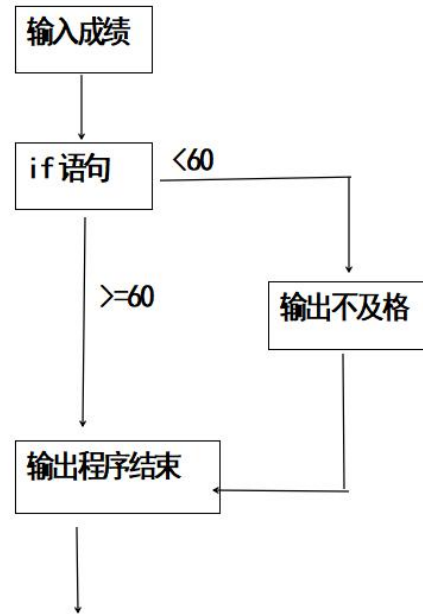
    if (i<60) {
        cout << "不及格" << endl;
    }
    cout << "程序结束" << endl;

    return 0;
}
```

1、输入34，贴运行结果

2、输入74，贴运行结果

3、画出程序对应的流程框图



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

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



§. 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
    }

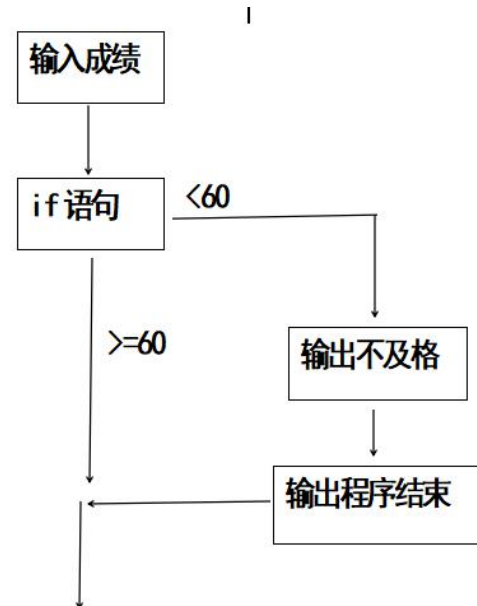
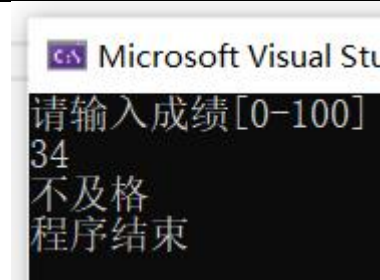
    return 0;
}
```

1、输入34，贴运行结果

2、输入74，贴运行结果

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

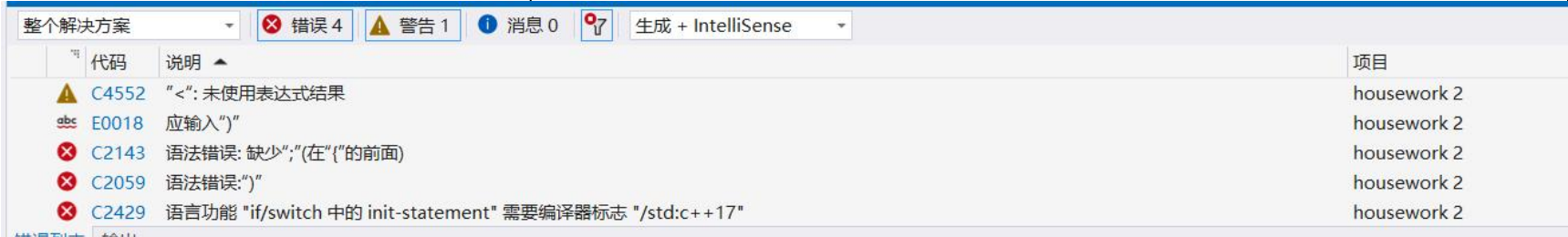




§. 基础知识题 - 关系运算、逻辑运算与选择结构

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)

<pre>#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; }</pre>	<div>贴编译错误并给出解释</div> <div></div> <div>if语句中使用的不是表达式，而是语句，是不符合语法规则的</div>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------



§. 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;
}
```

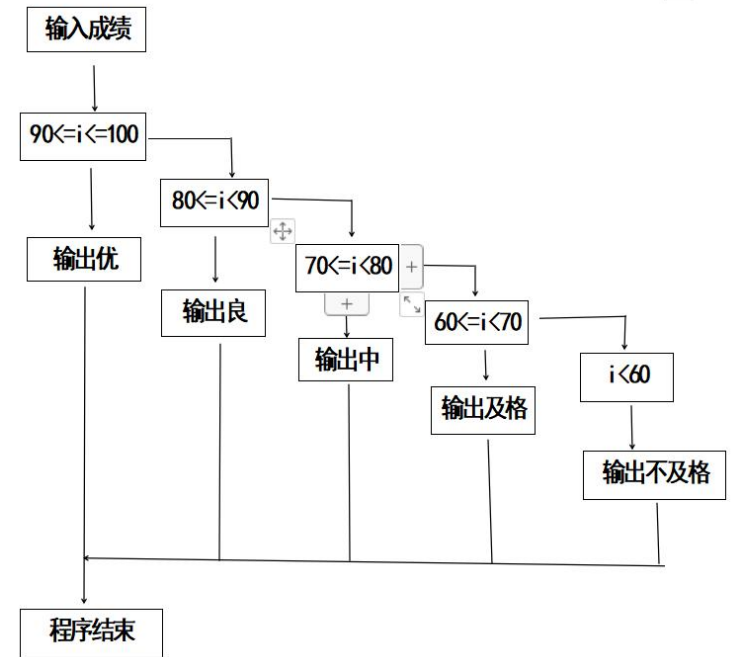
1、给出程序的流程框图(注意字体的清晰可辨)

2、 $i < 90$ 能否改为 $i \leq 89$? 哪个更好?

3、 $i < 90$ 能否改为 $i \leq 90$? 运行是否正确?

2. 可以，但是 $i < 90$ 更好，这样可以将数据划分更加合理，如果89.5也会输出良

3. 也可以，会先将90分给优，对输出的结果没有影响，但是并不好。



§. 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 看}, 总是与最近的{进行配对
-------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

§. 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____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

else总是与最近的if进行配对，上述式子被语句F中断，else没有与之配对的if语句。

错误列表		
整个解决方案		
错误 2 警告 0 展示 6 个消息中的 0 个		
代码	说明	
abc E0127	应输入一个语句	
✖ C2181	没有匹配 if 的非法 else	



§. 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>	左侧代码按缩进格式排版	<div>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</div> <div>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</div> <div>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</div>
<pre>if (表达式1) { if (表达式2) { A; } else { B; } C; }</pre>	左侧代码按缩进格式排版	<div>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</div> <div>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</div> <div>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</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



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

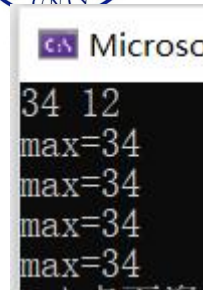
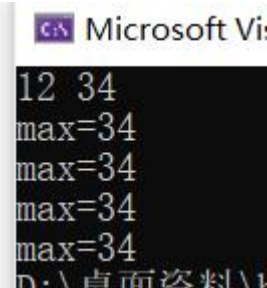
    printf("max=%d", a>b?a:b); //3

    return 0;
}
```

1、输入12 34，给出运行截图

2、输入34 12，给出运行截图

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

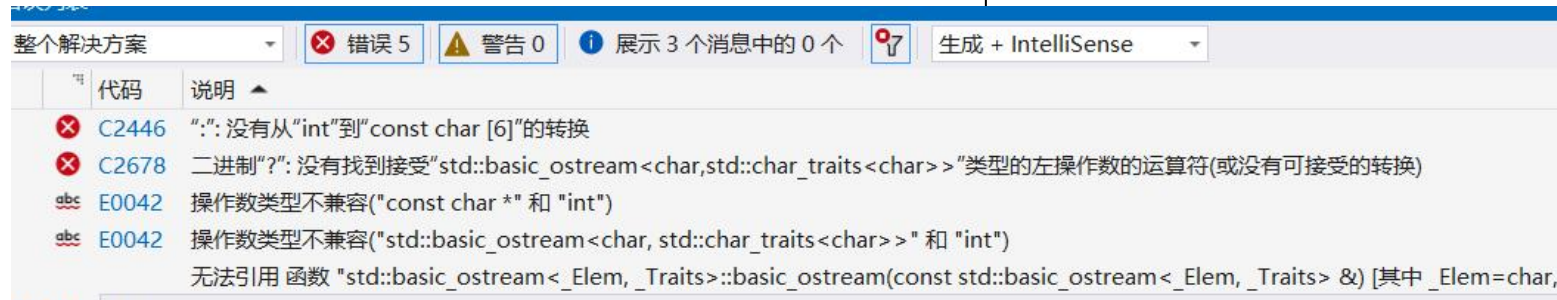
    a==1 ? 'A' : 123;              //编译正确

    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)

表达式一的类型可以和表达式二三的类型不同，但是表达式二三的类型必须相同，否则不能判断是值究竟是那种类型。



§. 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)

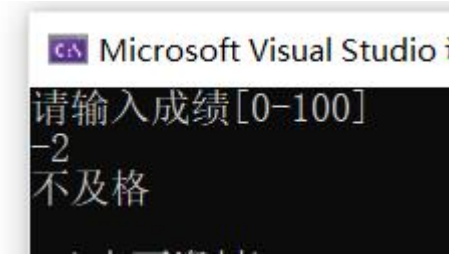
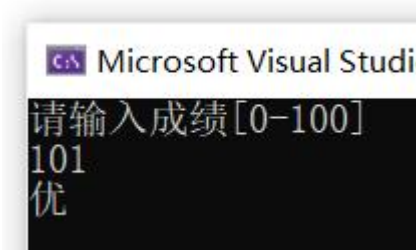
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 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

101和-2





§. 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)

是完全一致的，两者在错误上是相同的

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

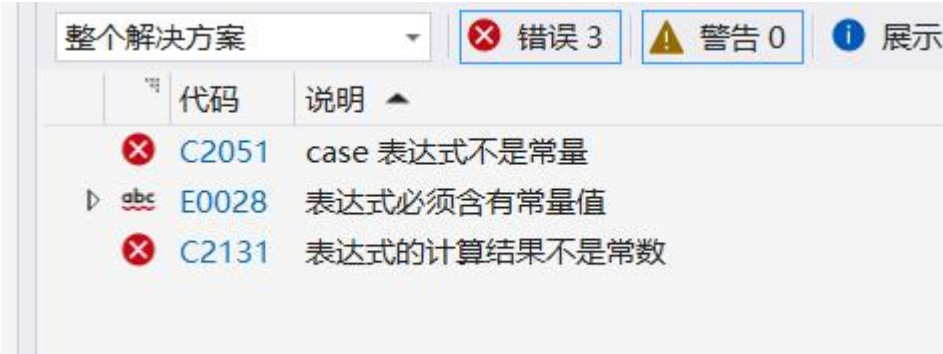
C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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
- 2、解释错误原因



将const int改为int后，表达式k+2就是一个变量加一个常量的形式，并不符合case语句的运算规则，是不行的。



§. 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)

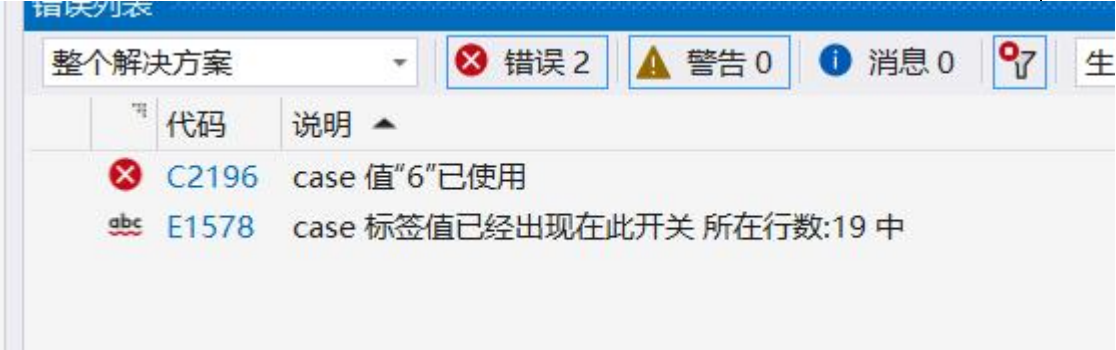
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case 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

2、解释错误原因



case 4+2 的形式与case 6的形式是一样的，在case表达式中同一个常数是不能连续用两次的。



§. 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

float的类型的值与int型的值进行乘除的运算的时候。其结果仍然是一个float型的数据，当进行case表达式的判断的时候，需要的是整数，所以发生了错误。


代码	说明	项目
abc E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	housework 2
abc E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	housework 2
abc E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	housework 2
abc E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	housework 2
abc E2373	此常量表达式的类型为 "int"，而所需类型为 "float"	housework 2



§. 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</p> <p>break语句是终止switch语句的条件，根据运行结果我们知道，没有break的语句的时候，会直接执行后面的语句，所以在输出良之后，又接着输出了一个中。</p> <div></div>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



§. 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，即符合预期的期望

Microsoft Visual Studio 调试

```
请输入成绩[0-100]
101
输入错误
```

Microsoft Visual Studio 调试控制

```
请输入成绩[0-100]
102
输入错误
```

Microsoft Visual Studio 调试

```
请输入成绩[0-100]
-1
输入错误
```

Microsoft Visual Studio 调试

```
请输入成绩[0-100]
-2
输入错误
```

```
#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:
            if(score<=100)
                cout << "优" << endl;
            else
                cout << "输入错误" << 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:
            if (score >=0)
                cout << "不及格" << endl;
            else
                cout << "输入错误" << endl;

            break;
        default:
            cout << "输入错误" << endl;
            break;
    }

    return 0;
}
```

§. 基础知识题 - 关系运算、逻辑运



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

2、如果用switch语句，该如何实现？
(部分即可)

3、如果学生成绩带小数点，即“xx.5”形式，能用switch语句吗？能用switch语句吗？
请解释原因

不能，Switch语句一般判断的是整数
可能需要强制转换

4、总结switch语句使用时的注意事项
一般用于整数的判断的题目，比较繁琐。

5、switch-case语句能完全取代if-els
不能

```
switch (score)
```

```
{
```

```
case 100:
```

```
case 99:
```

```
case 98:
```

```
case 97:
```

```
case 88:
```

```
case 87:
```

```
case 86:
```

```
case 85:
```

```
cout << "优" << endl;
```

```
case 4:
```

```
case 3:
```

```
case 2:
```

```
case 1:
```

```
case 0:
```

```
cout << "不及格" << endl;
```

```
default:
```

```
cout << "输入错误" << endl;
```

```
break;
```

```
case 84:
```

```
case 83:
```

```
case 82:
```

```
case 81:
```

```
case 74:
```

```
case 73:
```

```
case 72:
```

```
case 71:
```

```
case 70:
```

```
cout << "良" << endl;
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int score;
```

```
cout << "输入分数[0-100]" << endl;
```

```
cin >> score;
```

```
if (score <= 100 && score >= 85)
```

```
cout << "优" << endl;
```

```
else if (score >= 70 && score < 85)
```

```
cout << "良" << endl;
```

```
else if (score >= 60 && score < 70)
```

```
cout << "及格" << endl;
```

```
else if (score <= 60 && score >= 0)
```

```
cout << "不及格" << endl;
```

```
else
```

```
cout << "输入错误" << endl;
```

```
return 0;
```

```
}
```

```
case 69:
```

```
case 68:
```

```
case 67:
```

```
case 66:
```

```
case 65:
```

```
case 64:
```

```
case 63:
```

```
case 62:
```

```
case 61:
```

```
case 60:
```

```
cout << "及格" << endl;
```


§. 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