

§. 基础知识题 - 关系运算、逻辑运算与选择结构



要求:

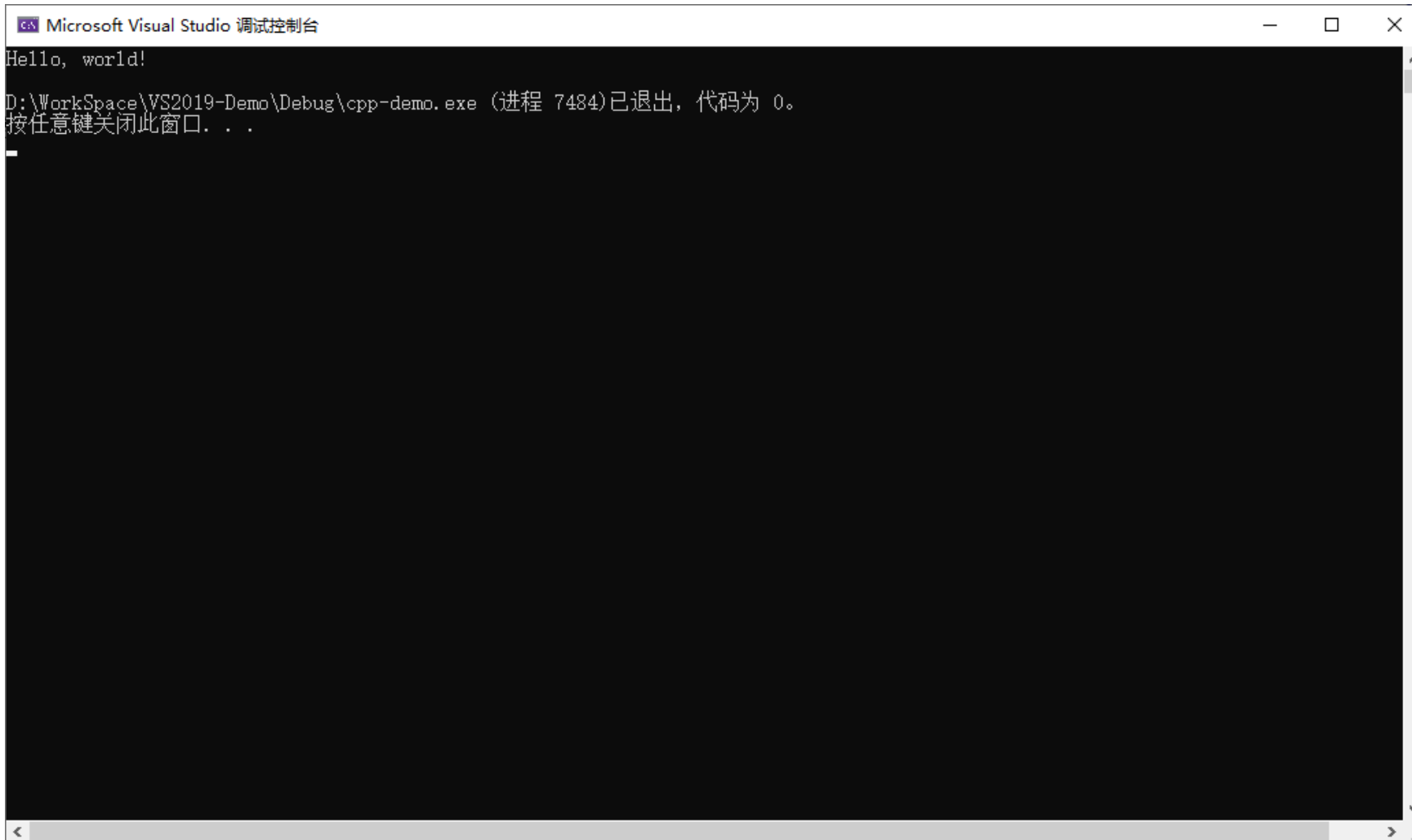
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**3月28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

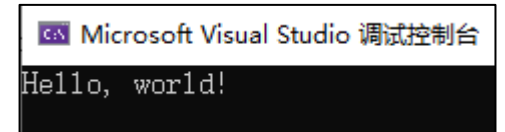
贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

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



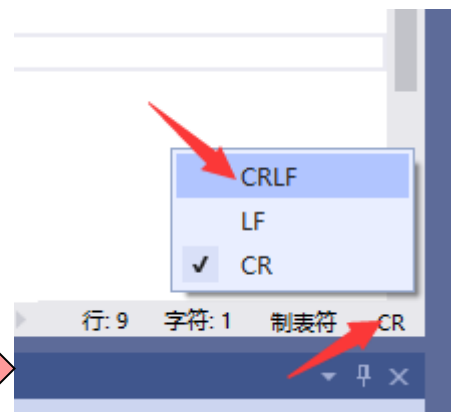
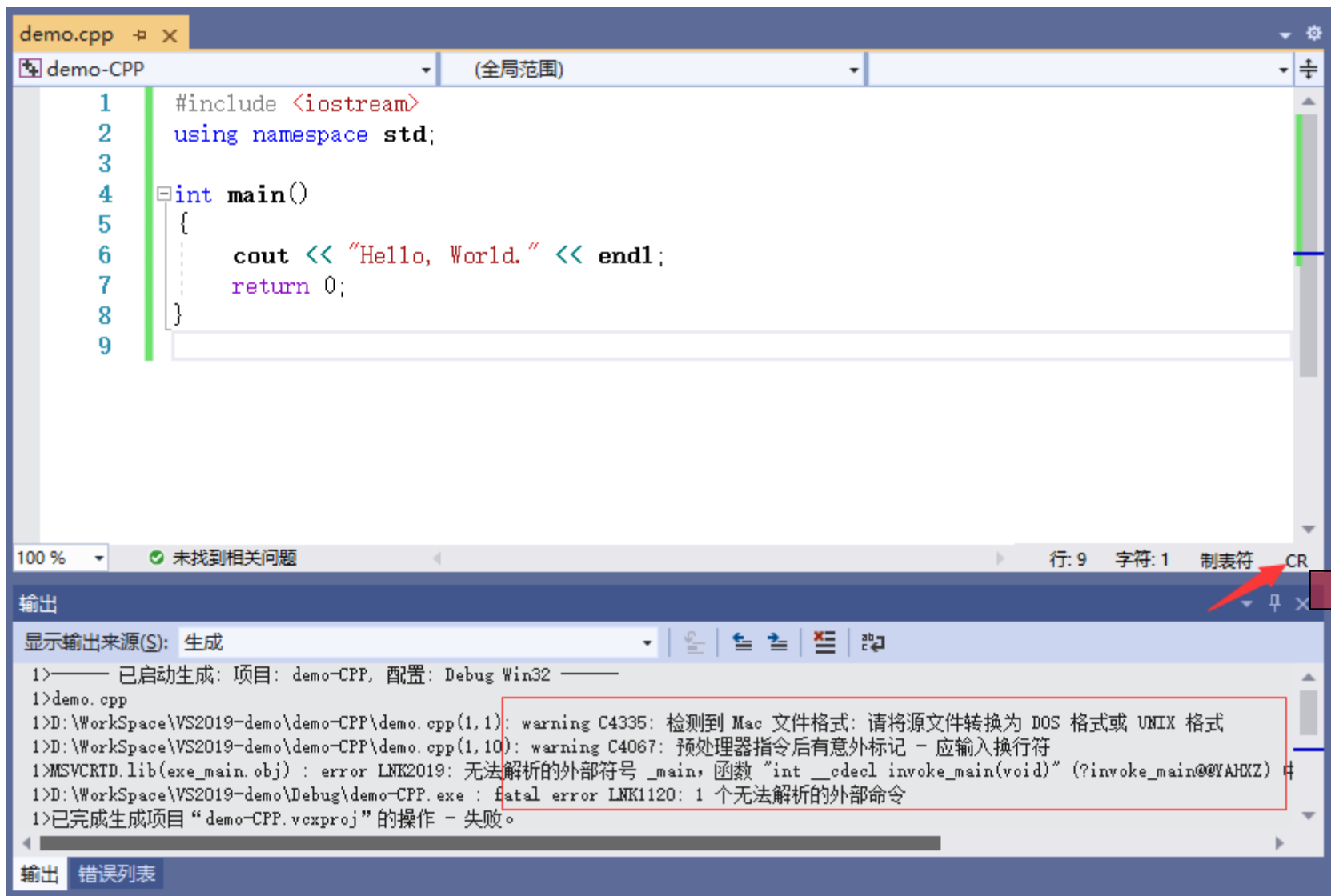
```
Microsoft Visual Studio 调试控制台
Hello, world!
```



§. 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



特别提示:

- ★ 本次作业的答案，除特别提示外，上课全讲过，课件上都有!!!
- ★ 作业本质就是对上课内容及课件的review(因为读懂程序的逻辑很重要)
- ★ 对上课接受程度较好的同学，可能有点重复/多余，但还得做



§. 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)


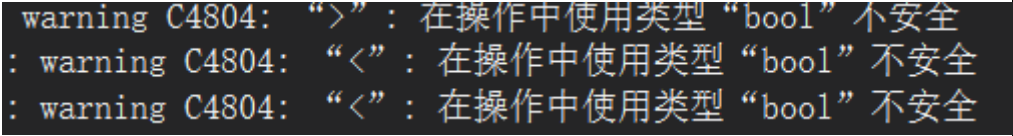
<pre>#include <iostream> using namespace std; int main() { int a=1, b=2, c=3, d; d = a > b > c; cout << d << endl; d = a < b < c; cout << d << endl; d = b > a < c; cout << d << endl; return 0; }</pre>	<div>0 1 1</div> <div>1、贴运行结果</div> <div>2、VS下为什么会有三个warning? 说说你的理解</div> <div>比较运算符左结合。 1. 表达式a>b即1>2为假，返回值为0，而0>3为假，所以值为0并赋值给d，d的输出结果为0; 2. 表达式a<b即1<2为真，返回值为1，而1<3为真，所以值为1并赋值给d，d的输出结果为1; 3. 表达式b>a即2>1为假，返回值为0，而0<3为真，所以值为1并赋值给d，d的输出结果为1。 出现warning代表使用了bool类型的值，在运算中被转换为整数值，并与数值进行比较，编译器给出warning。 计算机所理解的连续比较不同于人数学意义上的连续比较，所以会出现偏差，故编译器。</div>
---	--



§. 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></div> <div>2、$a > b > c$这个式子，按人的常规理解($3 > 2$且$2 > 1$)是正确的，为什么结果是0 ? $a < b < c$这个式子，按人的常规理解($3 < 2$且$2 < 1$)是错误的，为什么结果是1 ? $b > a < c$这个式子，按人的常规理解($2 > 3$且$3 < 1$)是错误的，为什么结果是1 ? (文字简单说明即可)</div> <div>1. 表达式$a > b$即$3 > 2$为真，返回值为1，而$1 > 1$为假，所以值为0并赋值给d，d的输出结果为0; 2. 表达式$a < b$即$3 < 2$为假，返回值为0，而$0 < 1$为真，所以值为1并赋值给d，d的输出结果为1; 3. 表达式$b > a$即$2 > 3$为假，返回值为0，而$0 < 1$为真，所以值为1并赋值给d，d的输出结果为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)

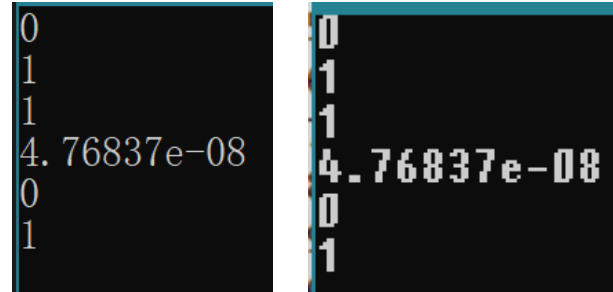
```
#include <iostream>
#include <cmath>
using namespace std;

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

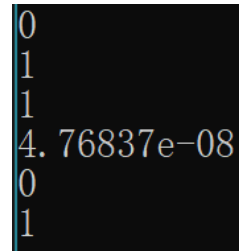


```
0
1
1
4.76837e-08
0
1

0
1
1
4.76837e-08
0
1
```

warning C4305: “初始化”：从“double”到“float”截断

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



```
0
1
1
4.76837e-08
0
1
```

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

warning C4305: “初始化”：从“double”到“float”截断

3、由本例得出的结论，实数进行相等比较时的通用方法是 将二数做差取绝对值（fabs函数），再与一极小值进行比较



§ . 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)

<pre>#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</pre>	<div>1、贴运行结果</div> <div></div> <div>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？</div> <div>在float下表现相同，在double下表现不同。 float的有效数字只有6位，而f1与f2前6位相同，所以fabs(f1-f2)为0，二值在float下完全相等。故0小于1e-6，小于1e-7均为真，返回1； double的有效数字有15位，精度不同。fabs(d1-d2)的值<1e-6，返回1；但>1e-7，返回0。</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

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和false是bool类型的bool常量，在内存中占一个字节，系统存储为是1和0.

"true" 和 "false" 是字符串，系统存储为字符串常量。

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？

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)

```
cout << boolalpha << true;
cout << boolalpha << 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(k2) << endl;
    cout << k2 << ' ' << int(k2) << endl;

    return 0;
}
```

1、贴运行结果

```
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__0/1__

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)

bool型常量/变量有两种：true和false。true为1，false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;

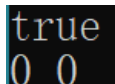
    cin >> k;
    cout << k << ' ' << int(k) << endl;

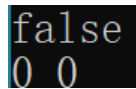
    return 0;
}
```

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型变量按“非0为真0为假”的原则。赋值非0，bool变量的值为1；赋值为0，则变量值为0.

使用cin输入时只能输入0/1，其余的均视为非法输入，返回值为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出(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

```
1 1
0 0
1 1
0
```

```
warning C4305: “=” : 从“char”到“bool”截断
: warning C4305: “=” : 从“int”到“bool”截断
2): warning C4305: “初始化” : 从“int”到“char”截断
4): warning C4309: “初始化” : 截断常量值
```

2、解释VS下waring的意思

bool变量只有1字节，将char型或int型数据赋值给bool型时会截断数据，造成数据丢失。

3、k='A' 是1字节赋值给1字节，为什么还有warning?

char虽然是1字节，但可以取不同的值；bool只有0/1两种值，所以发生了数据截断。

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？(和c对比)

$(256)_{10} = (1\ 00000000)_2$ ，截断后应该是0；实际为1；因为bool型变量遵循“非0为真0为假”的原则，所以被赋值为1.

5、为什么不 cout << c, 而是 (int)c ?

直接输出c会输出ASCII码下256所对应的字符，无法正常显示，转为int型后便于判断输出值。

6、“非0为真0为假”这句话如何解释?

bool型变量只有0/1而值，非0的值全都视为1；值为0时视为0. 1为真，0为假。



§. 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_____数字1/0_____



§. 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. $a>b$ $1>2$ 假为0

2. 0赋值给m $m=0$

3. 进行逻辑运算 $\&\&$ ，发现m为0，直接结束，n的值仍为1.

3、短路运算的意思是：

进行逻辑运算时

$\&\&$ ：一旦发现前面的值为0，则值必为0，停止运算，后面不求解

$\|\|$ ：一旦发现前面的值为1，则值必为1，停止运算，后面不求解



§. 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 a = 2, b = 2;
    cout << "a=" << a << " b=" << b << endl;
    5 > 3 && 2 || (a = 8 < 4 - !0), b = 8 < 4;
    cout << "a=" << a << " b=" << b << endl;
    return 0;
}
```

a=2 b=2
a=2 b=0

有以下逗号表达式，其表达式1是逻辑表达式，表达式2按需构造
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），证明两点：

- 1、8<4 - !0 存在短路运算
- 2、*** 不存在短路运算

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 比||优先级高的 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;
}
```

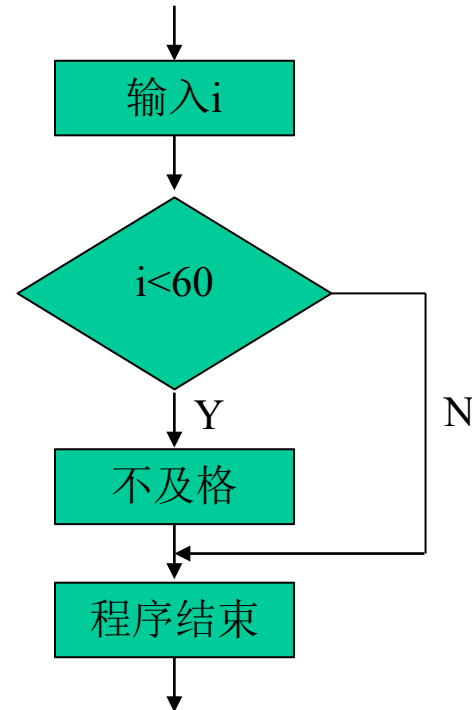
1、输入34，贴运行结果

```
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
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;
}
```

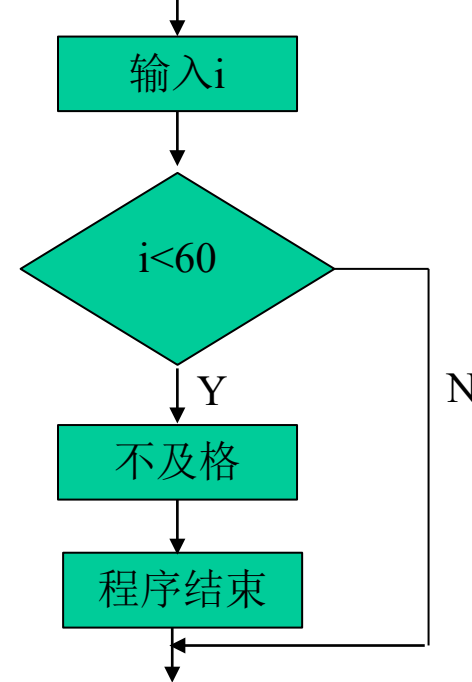
1、输入34，贴运行结果

```
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
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)

<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>warning C4552: " < ": 未使用表达式结果 error C2429: 语言功能 "if/switch 中的 init-statement" 需要编译器标志 "/std:c++17" error C2059: 语法错误: "(" error C2143: 语法错误: 缺少 ";" (在 "{" 的前面)</div> <div>if (表达式){ 语句序列; }</div> <div>() 内部应为表达式，此处为语句，故编译错误。</div>
--	--



§. 基础知识题 - 关系运算、逻辑运算与选择结构

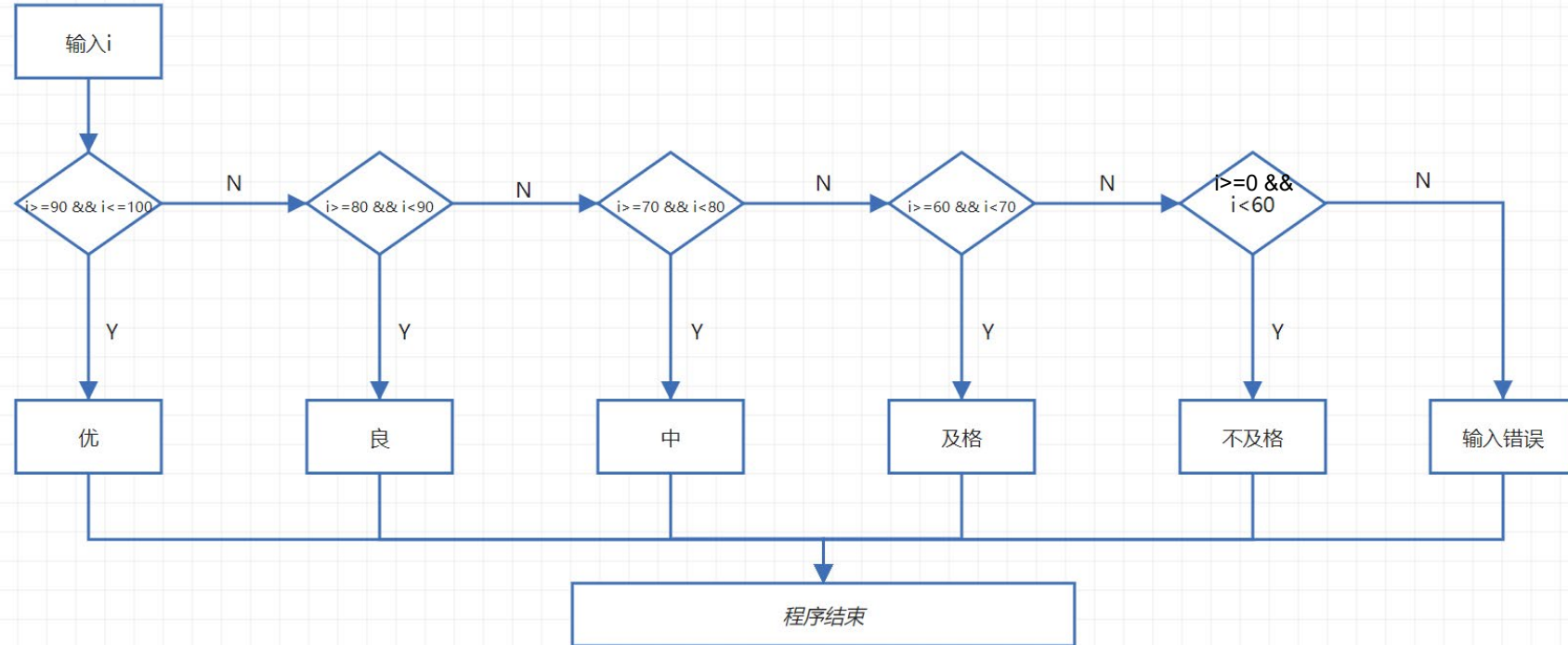
5、if语句 - 基本使用

D. 观察下列程序的运行结果，回答问题

```
#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?

可以。前者更好，表达的数据范围更为完整（如89.5）。

3、i<90能否改为i<=90? 运行是否正确?

可以。运行正确。90会被if后语句直接劫走，并不会执行else if后的语句。



§. 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，忽略{，以}为准向上匹配未配对的{。 以{}成对为基准，将else与它上面最近的if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 (1>2) {  
    if (3<4) {  
        cout << "case 1" << endl;  
    }  
    else {  
        cout << "case 2" << endl;  
    }  
    cout << "case 3" << endl;  
}  
cout << "case 4" << endl;  
else {  
    if (5>6) {  
        cout << "case 5" << endl;  
    }  
    cout << "case 6" << endl;;  
}  
return 0;
```

```
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

error C2181: 没有匹配 if 的非法 else

2、请说明错误原因

if-else双支语句中间不允许插入任何语句，插入语句会造成else语句单独存在，无法匹配与之成对的if，故编译器报错。



§. 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>	<ul style="list-style-type: none">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A2、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B3、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
<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>	<ul style="list-style-type: none">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A2、当表达式1__真__(真/假/任意)，表达式2__假__(真/假/任意)时，执行语句B3、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



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

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__3 2 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)

<pre>#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; }</pre>	<div data-bbox="1228 275 2265 354">1、给出编译报错的截图</div> <div data-bbox="1039 354 2489 492"><pre>error C2446: “:” : 没有从 “int” 到 “const char [6]” 的转换 message : 从整型类型转换为指针类型需要 reinterpret_cast、C 样式转换或带圆括号的函数样式强制转换 : error C2678: 二进制 “?” : 没有找到接受 “std::basic_ostream<char, std::char_traits<char>>” 类型的左操作数的运算符 (或没有可接受的转换) : message : 可以是 “内置 C++ operator?(int, int)” : message : 尝试匹配参数列表 “(std::basic_ostream<char, std::char_traits<char>>, int)” 时</pre></div> <div data-bbox="1228 492 2265 1360">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1 ? 表达式2 : 表达式3 条件运算符的表达式2，3不能为字符串，不能为返回值不是整数的表达式；只能接 字符常量，数字，返回值为整数的表达式 等且表达式2，3的类型必须相容，表达式1，2，3的类型可以不同。</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

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)

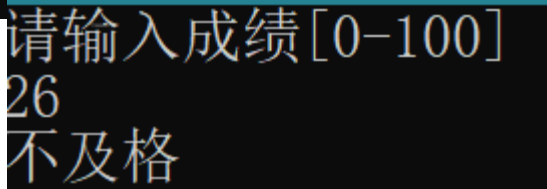
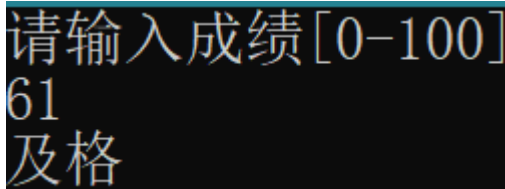
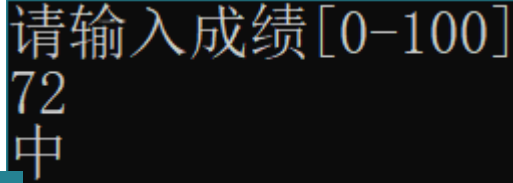
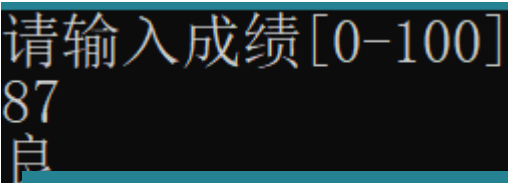
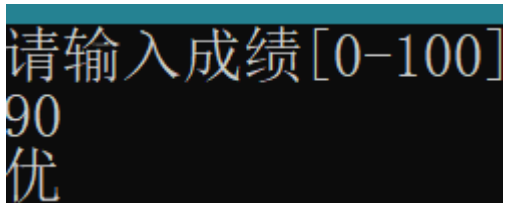
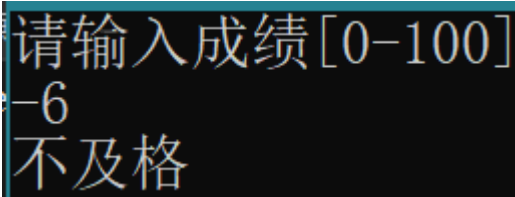
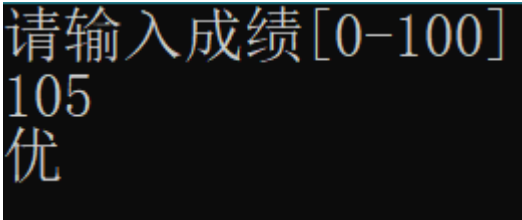
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case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

(-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)

```
#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)

请输入成绩[0-100]
105
优

请输入成绩[0-100]
-6
不及格

请输入成绩[0-100]
90
优

请输入成绩[0-100]
87
良

请输入成绩[0-100]
72
中

请输入成绩[0-100]
61
及格

请输入成绩[0-100]
26
不及格

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)

<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> <div><p>1、给出编译错误的截图</p><p>2、解释错误原因</p><p>case后面的表达式必须是字符常量或整型常量，不能是变量。</p></div> <div><pre>error C2131: 表达式的计算结果不是常数 message : 因读取超过生命周期的变量而失败 message : 请参见“k”的用法 error C2051: case 表达式不是常量</pre></div>
--	--



§. 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

error C2196: case 值“6”已使用

2、解释错误原因

case后的值不能重复出现



§. 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)

<pre>#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; }</pre>	<p>在8.A的基础上，将score从int改为float</p> <p>1、给出编译错误的截图</p> <p>2、解释错误原因</p> <p>switch语句()内只能是字符常量或整型常量，不能为浮点型常量float。</p>
---	--

error C2450: 类型为“float”的 switch 表达式无效
message : 要求整型表达式



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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;

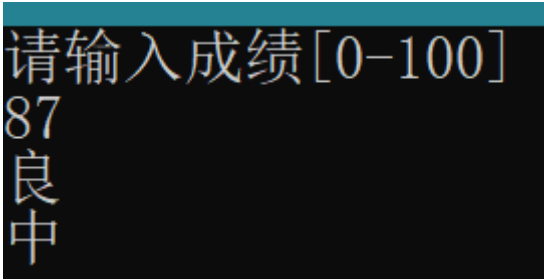
        case 7:
            cout<<"中"<<endl;
            break;
        case 6:
            cout<<"及格"<<endl;
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<"不及格"<<endl;
            break;
        default:
            cout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

使程序跳出循环，不再执行后续语句；否则连续执行下一case语句。





§. 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;

        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 < 110 || score > -10 && score < 0)
        cout << "输入错误" << endl;
    else {
        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：
[84-100] - 优
[68-84) - 良
[55-68) - 及格
[0-55) - 不及格

1、用if-else语句完成该程序并贴图 见下页

2、如果用switch语句，该如何实现？ 见下页（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
可以用if，不可以用switch。

if对数据无限制，但switch限制数据为整型。

4、总结switch语句使用时的注意事项

switch语句()内只能为整型表达式，不可为浮点型、小数等；

case后的值只能为常量；

case后需跟break，否则程序会一直往下进行；

case后面的值不能重复，也不能有矛盾。

5、switch-case语句能完全取代if-else吗？

不能。switch使用具有局限性，但适合表示多分支语句。



§. 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

```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout << "请输入成绩[0-100]" << endl;
    cin >> score;
    if (score > 100 && score < 110 || score > -10 && score < 0)
        cout << "输入错误" << endl;
    else {
        if (score >= 84 && score <= 100) {
            cout << "优" << endl;
        }
        else if (score >= 68 && score < 84) {
            cout << "良" << endl;
        }
        else if (score >= 55 && score < 68) {
            cout << "及格" << endl;
        }
        else if (score >= 0 && score < 55) {
            cout << "不及格" << endl;
        }
        else {
            cout << "输入错误" << endl;
        }
    }
    return 0;
}
```

```
switch (score) {
    case 100:
    case 99:
    case 98:
    case 97:
    case 96:
    case 95:
    case 94:
    case 93:
    case 92:
    case 91:
    case 90:
    case 89:
    case 88:
    case 87:
    case 86:
    case 85:
    case 84:
        cout << "优" << endl;
        break;
    case 83:
    case 82:
    case 81:
    case 80:
    case 79:
    case 78:
    case 77:
    case 76:
    case 75:
    case 74:
    case 73:
    case 72:
    case 71:
    case 70:
    case 69:
    case 68:
        cout << "良" << endl;
        break;
    case 67:
    case 66:
    case 65:
    case 64:
    case 63:
    case 62:
    case 61:
    case 60:
    case 59:
    case 58:
    case 57:
    case 56:
    case 55:
        cout << "及格" << endl;
        break;
    default:
        cout << "输入错误" << endl;
        break;
}
```



§. 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