

§ . 基础知识题 - 关系运算、逻辑运算与选择结构



要求:

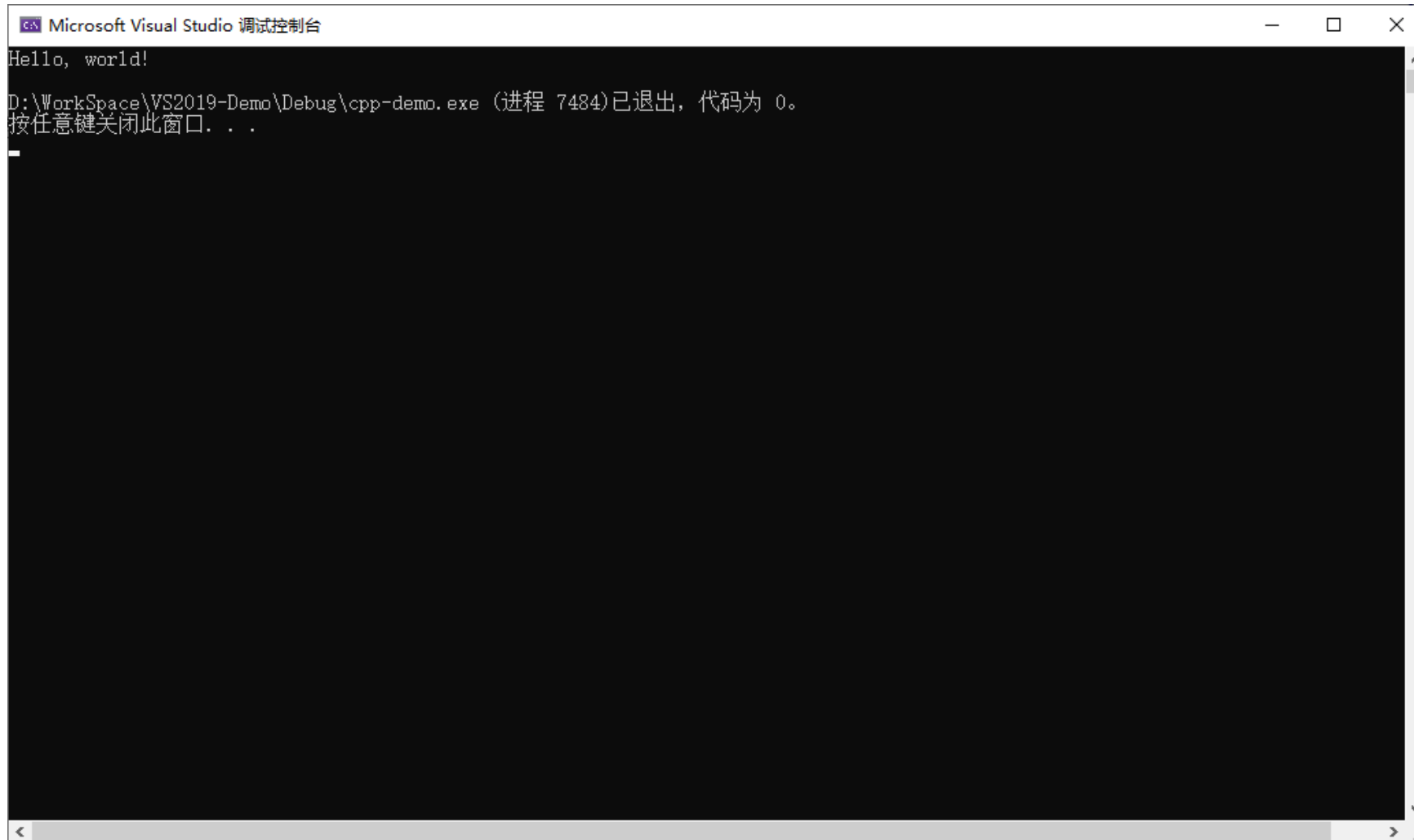
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果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

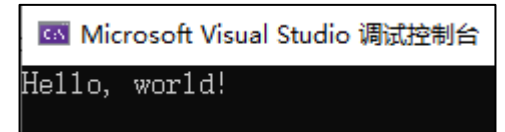
贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

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



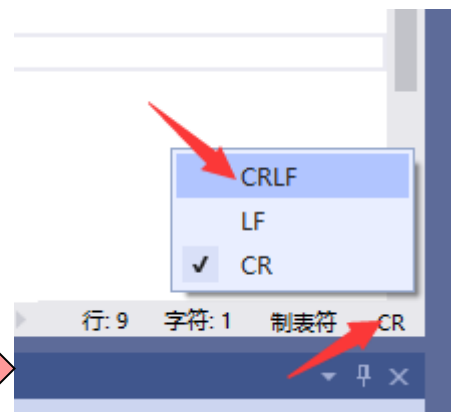
```
Microsoft Visual Studio 调试控制台
Hello, world!
```



§. 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

Microsoft Visual

```
0
1
1
```

1>源.cpp

```
1>C:\Users\lei\Desktop\Project\Test\源.cpp(8,11): warning C4804: ">" : 在操作中使用类型 "bool" 不安全
1>C:\Users\lei\Desktop\Project\Test\源.cpp(11,11): warning C4804: "<" : 在操作中使用类型 "bool" 不安全
1>C:\Users\lei\Desktop\Project\Test\源.cpp(14,11): warning C4804: "<" : 在操作中使用类型 "bool" 不安全
```

2、VS下为什么会有三个warning? 说说你的理解

这三个warning代表同一种错误，以 $d = a > b > c$ 为例，我们的想法是b小于a且b大于c，而编译器不是这样理解的，编译器认为 $d = (a > b) > c$ ，即编译器先判断 $(a > b)$ 返回一个布尔值，0或1，只有这两种情况，再将0或1和c进行比较。

如果要实现我们的原本想法，应使用 $d = (a > b) \&\& (b > c)$;



§. 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



```
1>源.cpp
1>C:\Users\lei\Desktop\Project\Test\源.cpp(8,11): warning C4804: ">" : 在操作中使用类型 "bool" 不安全
1>C:\Users\lei\Desktop\Project\Test\源.cpp(11,11): warning C4804: "<" : 在操作中使用类型 "bool" 不安全
1>C:\Users\lei\Desktop\Project\Test\源.cpp(14,11): warning C4804: "<" : 在操作中使用类型 "bool" 不安全
```

- 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？

$a > b > c$ 编译器先进行 $(a > b) > c$ ，因为 $a=3$ ， $b=2$ ，所以 $(a > b)$ 返回1，而 $c=1$ ，1不大于1，所以结果是false，即0；

$a < b < c$ 编译器先进行 $(a < b) < c$ ，因为 $a=3$ ， $b=2$ ，所以 $(a < b)$ 返回0，而 $c=1$ ，0小于1，所以结果是true，即1；

$b > a < c$ 编译器先进行 $(b > a) < c$ ，因为 $a=3$ ， $b=2$ ，所以 $(b > a)$ 返回0，而 $c=1$ ，0小于1，所以结果是true，即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)

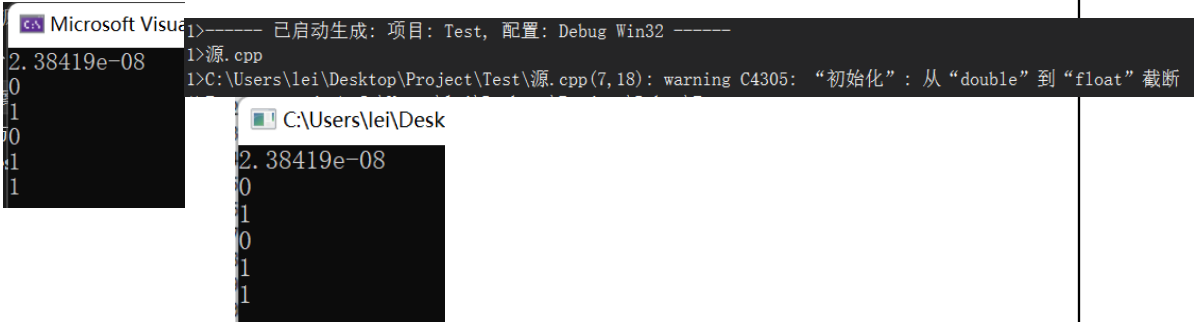
```
#include <iostream>
#include <cmath>
using namespace std;

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

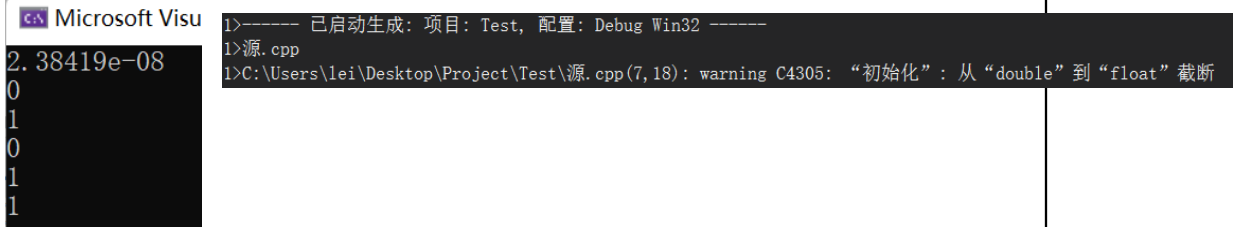
    float c = 1.0;
    cout << (c - 1.0) << endl;
    cout << (c == 1.0) << endl;
    cout << (fabs(c - 1.0) < 1e-6) << endl;

    return 0;
}
```

1、贴VS+Dev下的运行结果



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



行	列	单元	信息
10	14	C:\Users\lei\Desktop\Project\Test\源.cpp	In function 'int main()':
		C:\Users\lei\Desktop\Project\Test\源.cpp	[Error] 'fabs' was not declared in this scope; did you mean 'labs'?

3、由本例得出的结论，实数进行相等比较时的通用方法是__比较两个实数差的绝对值是否小于一个极小数，比如是否小于1e-6__



§. 基础知识题 - 关系运算、逻辑运算与选择结构

2、关系运算符与实数

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

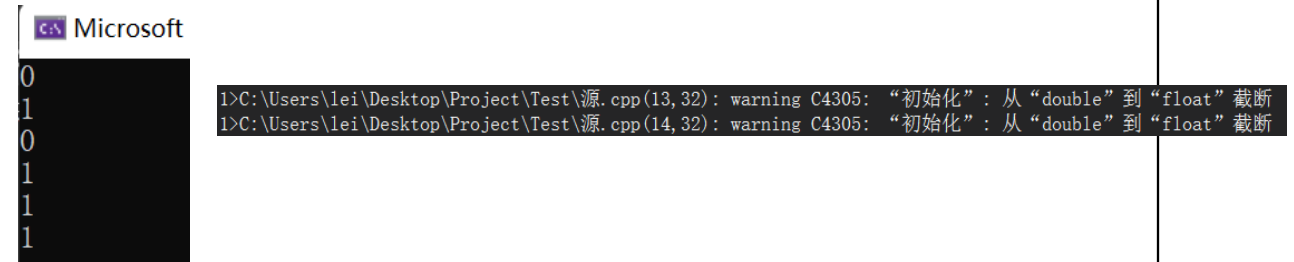
```
#include <iostream>
#include <cmath> //VS可不加
using namespace std;

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

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

    return 0;
} //VS有两个warning
```

1、贴运行结果



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

$\text{fab}(**)<1e-6$ 和 $\text{fabs}(**)<1e-7$ 在float下表现相同，在double下不相同。Float有效数字有6位，当f1和f2前六位小数均相同时， $f1-f2$ 得到结果是0.00000000，小于 $1e-6$ 和 $1e-7$ ；而double有效数字有15位，当d1和d2前六位小数均相同时， $d1-d2$ 得到的结果是-0.00000011，其绝对值小于 $1e-6$ ，但大于 $1e-7$



§. 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"true")

true/false是布尔值； "true" / "false" 是字符串

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

```
cout << boolalpha << true << endl;
```

加入前导格式控制符boolalpha



§. 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;

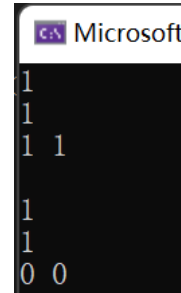
    cout << sizeof(true) << endl;
    cout << sizeof(k1) << endl;
    cout << k1 << ' ' << int(k1) << endl;

    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)

bool型常量/变量在输出时，true对应输出1，
false对应输出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)

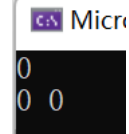
```
#include <iostream>
using namespace std;

int main()
{
    bool k;

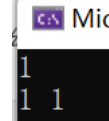
    cin >> k;
    cout << k << ' ' << int(k) << endl;

    return 0;
}
```

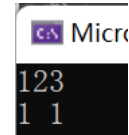
1、输入0，输出是：



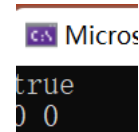
2、输入1，输出是：



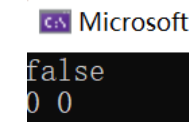
3、输入123，输出是：



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型变量在用cin进行输入时，只有0/1为正确输入，其他均非法。在遇到非0非1等非法输入时，若是非0非1的数字，则返回1；若是字母等其他字符，则返回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;
}
```

在进行赋值操作时，将非0值赋值给bool型变量，变量为true，将0赋值给变量，变量为false

1、贴运行结果

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

```
1>源.cpp
1>C:\Users\lei\Desktop\Project\Test\源.cpp(7,12): warning C4305: "=" : 从"char"到"bool"截断
1>C:\Users\lei\Desktop\Project\Test\源.cpp(13,12): warning C4305: "=" : 从"int"到"bool"截断
```

2、解释VS下warning的意思

第一个warning对应 `k='A'`；'A' 是一个char型字符，而k是一个bool型变量，两者类型不同

第二个warning对应 `k=256`；256是一个int型常量，而k是一个bool型变量，且int型字节大于bool型字节，所以会截断

3、`k='A'` 是1字节赋值给1字节，为什么还有warning?

bool虽然占一个字节，但实际有效位数为1位，即只有最低位有效，存储0或1，其他7位均恒为0

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?

k为0, 256 int型对应内存存储
00000000 00000000 00000001 00000000, 截断后为00000000

5、"非0为真0为假"这句话如何解释?



§. 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)

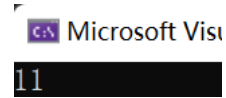
```
#include <iostream>
using namespace std;

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__
true按1计算，false按0计算__



§. 基础知识题 – 关系运算、逻辑运算与选择结构

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

A. 完成下列两个表格的填写（a/b是两个逻辑值，填写的内容不要用黑色）

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

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



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

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

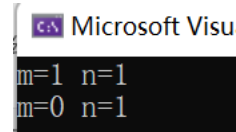
```
#include <iostream>
using namespace std;

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

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

    return 0;
}
```

1、贴运行结果



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

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

- ① $a>b$ 先计算 $a>b$, 得false,
- ② $m=a>b$ 将false赋值给m, 即 $m=0$
- ③ 表达式 $(m=a>b)\&\&(n=c>d)$ 的值为0

3、短路运算的意思是: _

对于 $\&\&$, 若左边表达式的值为0, 则不进行右边表达式的计算, 整个表达式的值为0; 对 $\|\|$, 若左边表达式的值为1, 则不进行右边表达式的计算, 整个表达式的值为1



§. 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;
```

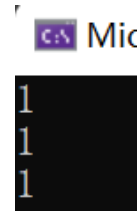
```
int main()
{
    cout << (5 > 3 && 2 || 8 < 4 - !0) << endl;
    cout << (5 > 3 && 2 || 1) << endl;
    cout << (5 > 3 && 2 || 0) << endl;
    return 0;
}
```

有以下逻辑表达式

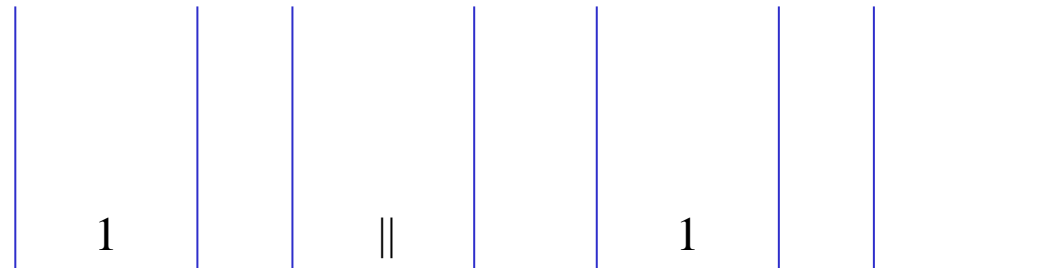
$5 > 3 \ \&\& \ 2 \ || \ 8 < 4 - !0$

- 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

$8 < 4 - !0$ 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）



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

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;
}
```

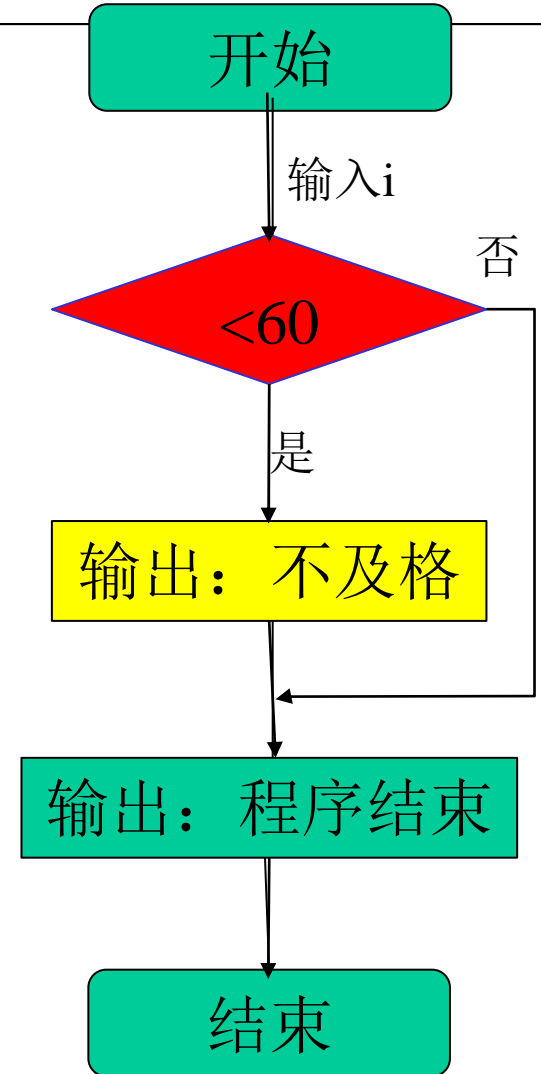
1、输入34，贴运行结果

```
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;

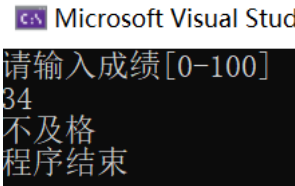
int main()
{
    int i;

    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

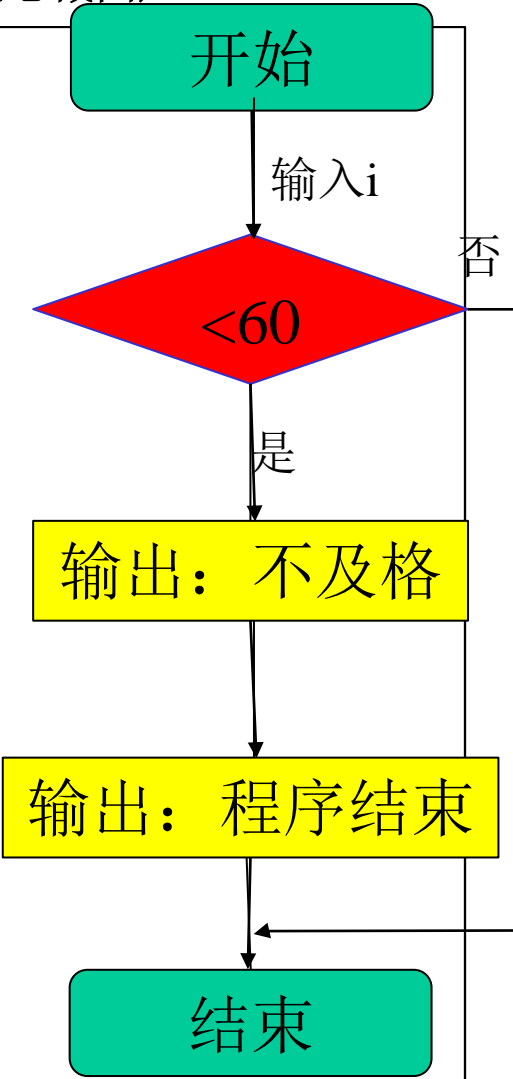


2、输入74，贴运行结果



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)

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

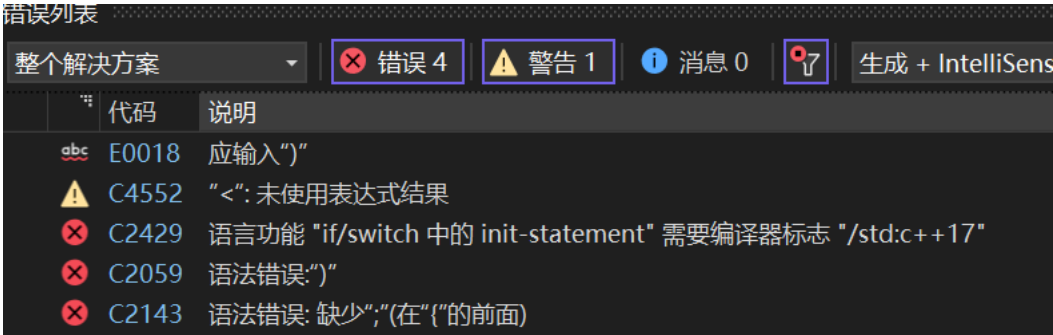
int main()
{
    int i;

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

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

    return 0;
}
```

贴编译错误并给出解释



if (表达式)
{
 语句
}
if判断语句应该填写一个表达式，
而不能填写语句
该程序(i<60;)作为一个语句作为if的
判断语句，故报错



§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

D. 观察下列程序的运行结果，回答问题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;
}
```

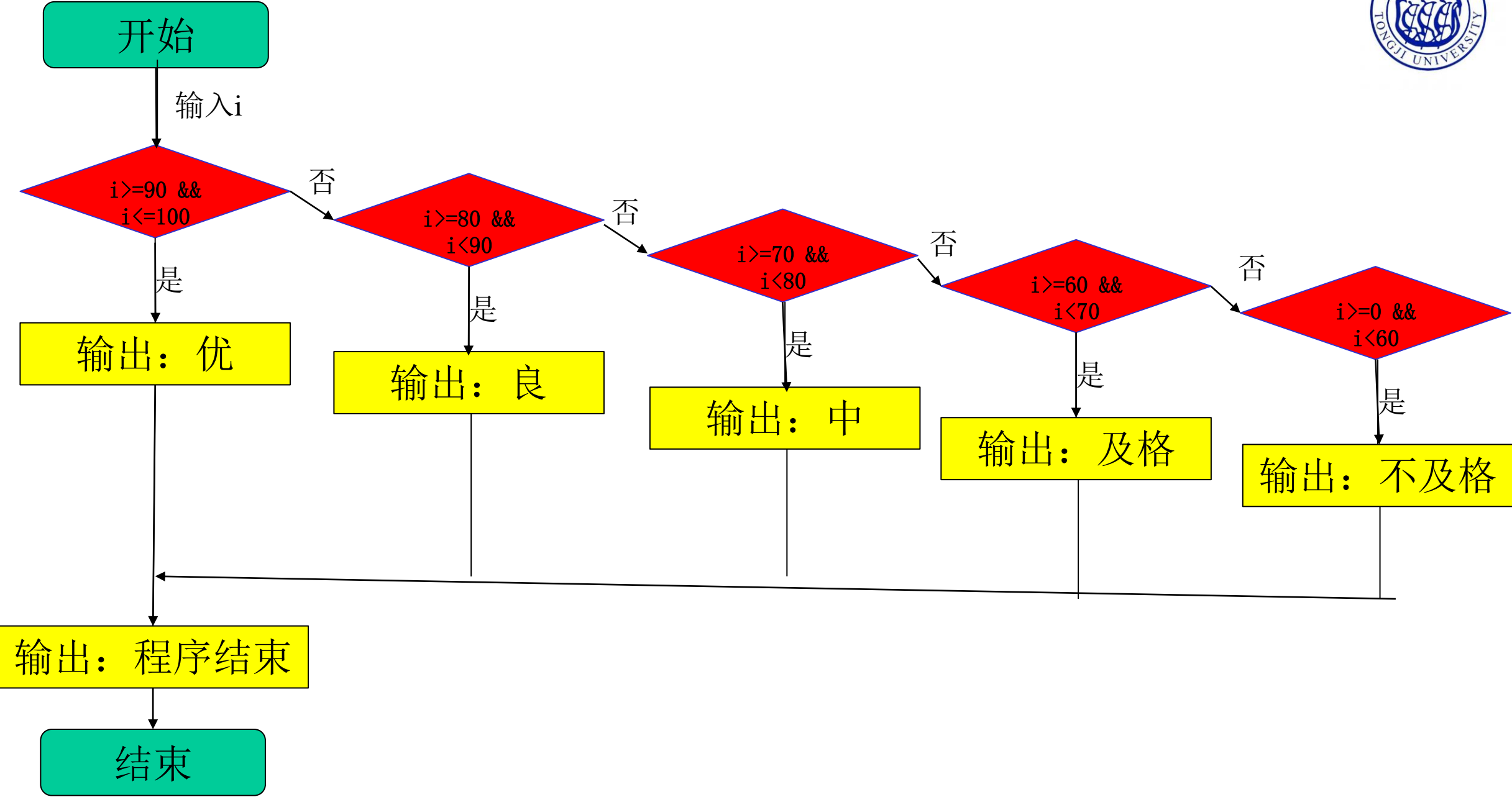
1、给出程序的流程框图(注意字体的清晰可辨)

2、 $i < 90$ 能否改为 $i \leq 89$? 哪个更好?

$i < 90$ 更好。

使用 $i \leq 89$ 时，编译器需要进行两步操作，
先判断是否小于89，再判断是否等于89，
而使用 $i < 89$ 只需要进行一步判断

3、 $i < 90$ 能否改为 $i \leq 90$? 运行是否正确?
正确





§. 基础知识题 – 关系运算、逻辑运算与选择结构

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 – 关系运算、逻辑运算与选择结构

6、if语句 – 多重嵌套

A. 一个有10行代码的if语句嵌套，回答问题

0: if (表达式) { 1: if (表达式) { 2: } 3: else { 4: } 5: } 6: else { 7: if (表达式) { 8: } 9: }	第0行的"{" 和 第_5_行的"}"配对 第1行的"{" 和 第_2_行的"}"配对 第3行的"{" 和 第_4_行的"}"配对 第6行的"{" 和 第__9_行的"}"配对 第7行的"{" 和 第_8_行的"}"配对 总结：给出大括号配对的基本准则 自上而下，忽略{，以}为准向上匹配最近的未匹配过的}
---	---



§. 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

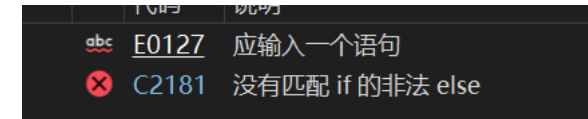
D. 一个if语句嵌套如下，回答问题

```
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

```
int main()  
{  
    int a = 1, b = 2, c = 3, d = 4;  
    if (a==b)  
    {  
        if (b==c)  
        {  
            cout << "A" << endl;  
        }  
        else  
        {  
            cout << "B" << endl;  
        }  
        cout << "C" << endl;  
    }  
    cout << "F" << endl;  
    else  
    {  
        if (c==d)  
        {  
            cout << "D" << endl;  
        }  
        cout << "E" << endl;  
    }  
}
```



2、请说明错误原因

编译器自上而下运行，当运行到
cout << "F" << endl;时，认为if
语句已结束，即上面的if为单分支
结构，而cout下面又出现了else，
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
12 34
max=34
max=34
max=34
max=34
max=34
```

2、输入34 12，给出运行截图

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

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

形如“表达式1? 表达式2:表达式3”的条件表达式
表达式1、2、3的类型可以不同，但2、3的类型比如相容，所谓相容，比如char和int就可以相容



§. 基础知识题 – 关系运算、逻辑运算与选择结构

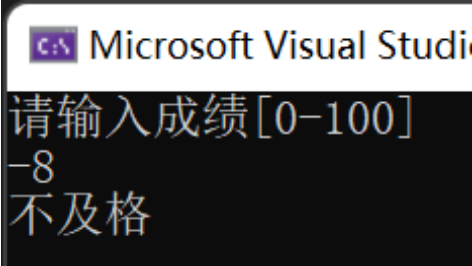
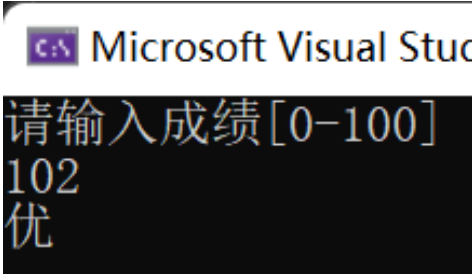
此页不要删除，也没有意义，仅仅为了分隔题目



§. 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)

<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”</p> <p>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</p> <p>输入 (-10, 0) 时，返回不及格，正确结果应该是输入错误； 输入 (100, 110) 时，返回优，正确结果应该是输入错误</p> <div></div> <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

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)

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

    return 0;
}
```

在8.B的基础上，将k从const int改为int

1、给出编译错误的截图



2、解释错误原因

case值只能是常数，不能是变量或含变量的表达式

此处k+2就是一个含变量的表达式



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

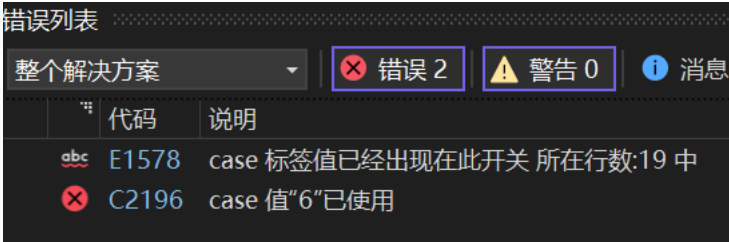
D. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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值不能出现重复，此处4+2的值为6，而case值已经存在等于6这种情况，所以报错



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

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

    return 0;
}
```

在8.A的基础上，将score从int改为float

1、给出编译错误的截图



2、解释错误原因

score为float型变量，switch的表达式score/10得到的结果也是一个float型数据，而该处只能接受整型结果，所以报错



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

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

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

    return 0;
}
```

在8.A的基础上，删除case 8后面的break

1、给出与8.A运行结果不一致的测试数据即截图



2、解释break的作用

执行完相应case后跳出switch语句。如果没有break，在执行完满足条件的case后，会继续执行后续case，直到遇到break或default才结束switch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

```
#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

```
#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：

[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

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

可以使用if语句，也可以使用switch语句。If语句可以判断xx.5和整数的大小关系，switch语句可以借助int进行强制转化，虽然说比较麻烦，勉强可以使用

4、总结switch语句使用时的注意事项

- ① case值只能为整型或char型，不能有浮点数；
- ② 表达式的结果只能是整形；
- ③ 不能有相同的case值；
- ④ case中注意break的使用

5、switch-case语句能完全取代if-else吗？

不能，switch-case语句主要还是用来处理对单条语句并且结果为整型的条件判断上，而if-else使用范围要广得多，不局限于特定类型

```
#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 << "输入错误" << endl;
    }
    else if (score >= 85 && score <= 100)
    {
        cout << "优" << endl;
    }
    else if (score >= 70 && score < 85)
    {
        cout << "良" << endl;
    }
    else if (score >= 60 && score < 70)
    {
        cout << "及格" << endl;
    }
    else
    {
        cout << "不及格" << endl;
    }
    return 0;
}
```



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）

```
if (score < 0 || score > 100)
{
    cout << "输出错误" << endl;
}
else
{
    switch (score / 10)
    {
        case 10:
        case 9:
            cout << "优" << endl;
            break;
        case 7:
            cout << "良" << 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;
    }
    switch ((score + 5) / 10)
    {
        case 9:
            cout << "优" << endl;
            break;
        case 8:
            cout << "良" << endl;
            break;
    }
}
```

总体思路：利用两个switch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