

§. 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

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

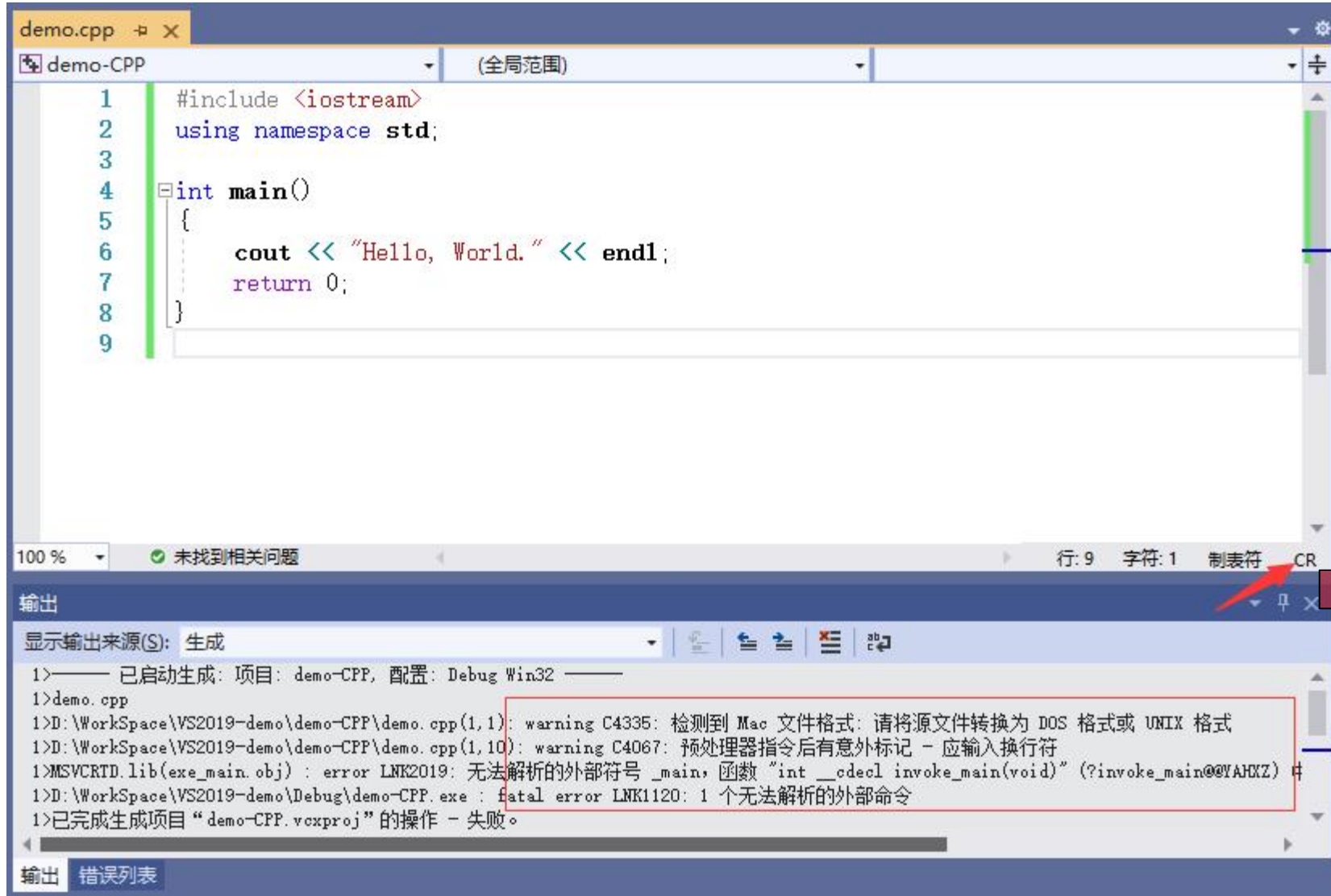
A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the output text. The window is titled "Microsoft Visual Studio 调试控制台". It contains the text: "Hello, world!". This is an example of an effective screenshot that focuses on the relevant information.



§. 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)

```
#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

当你在vs中写这样的比如第一句a>b>c; 实际上执行的是a>b得到的是假的bool这时候是0, 然后在判断0>c也是假, 值为0。因此输出d是0; 同理下面两句也是。当vs检测到这样的语法的时候就警告提示你这里不是这样运行的



§. 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



⚠ C4804	"<": 在操作中使用类型"bool"不安全	cpp	源.cpp	11
⚠ C4804	"<": 在操作中使用类型"bool"不安全	cpp	源.cpp	14
⚠ C4804	">": 在操作中使用类型"bool"不安全	cpp	源.cpp	8

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)

- 1: 其实是 $a > b$ 然后得到true值为1，再判断 $1 > c$ 是false错的因此值为0
- 2: 其实是 $a < b$ 然后的到false值为0，再判断 $0 < c$ 是true对的值为1
- 3: 其实是 $b > a$ 是错的得到false为0，再判断 $0 < c$ 是true对的值为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 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



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

使用==判断符号；而不是使用库函数，如果去掉头文件可能会报错；也不是使用两个实数相减，如果是浮点型数字出现储存误差可能会得不到想要的结果



§. 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)

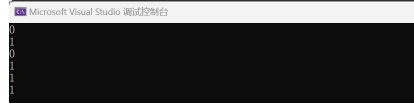
```
#include <iostream>
#include <cmath> //VS可不加
using namespace std;

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？

小于1e-6相同而小于1e-7不同；
因为float型数据有效位数是6，f1和f2在前6位相同，相减直接的到0, 0自然小于1e-6和1e-7两个值

而double型数据的有效位数是15，f1和f2相减之后小于1e-6但是会大于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;

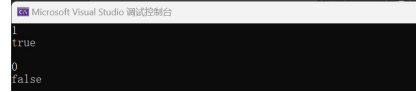
int main()
{
    cout << true    << endl;
    cout << "true"  << endl;

    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"false")

true和false是bool型类型的数据
而 "true"和"false"是字符串

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)

可以

```
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



2、bool型常量/变量在内存中占用__1__字节，值是__true是1、false是0__

总结bool型常量/变量在输出时的规则
(限制：在无3.A的前导格式控制符的前提下)

输出true时就会输出1;
输出false时就会输出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：
输入任何非零数的时候都计算为1，也就是true；
在输入0的时候计算为0，也就是false；
而输入非法字符比如字符的时候，都计算为0，也就是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

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;

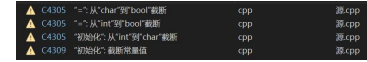
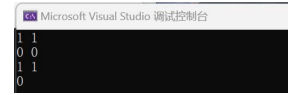
    k=0;
    cout << k << ' ' << (int)k << endl;

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

    char c = 256;
    cout << (int)c << endl;

    return 0;
}
```

1、贴运行结果



2、解释VS下warning的意思

第三和第四个warning都是对int传输给char值可能发生的数据截断丢失；第二个是对int传输给bool值0/1可能得截断丢失；第一个是对char值赋值给bool发生的数据截断丢失

3、k='A' 是1字节赋值给1字节，为什么还有warning？
char字符是以ascii码值存贮，而bool值只有1和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？(和c对比)

k应该是接受了256的补码然后转化为有符号数的-127；
现在实际是1；因为bool实际上应该只接受第一位有效位数也就是1

5、为什么不 cout << c，而是 (int)c？

cout<<c就会输出对应ascii码对应的字符了

6、“非0为真0为假”这句话如何解释？

对于bool型数据来说，任何非零的数传给bool型都是真；而只有0传给bool型才是假



§. 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____数字的1或者是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

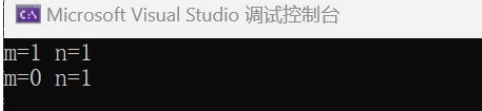
a	b	!a	!b	a&&b	a b
非0	非0	0	0	1	1
非0	0	0	1	0	1
0	非0	1	0	0	1
0	0	1	1	0	0



§. 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)

<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>2、解释 (m=a>b)&&(n=c>d) 的求值过程（标出步骤顺序）</div> <div>1: a>b为假得0</div> <div>2: 然后m=0</div> <div>3: 返回一个0</div> <div>4: 短路运算发现已经可以确定是假因此直接输出0，右侧括号没有计算</div> <div>3、短路运算的意思是：___在顺序进行计算时如果能直接确认结果的话，剩下的就可以跳过不计算___</div> <div>对&&来说，如果左侧结果是false或者0；就直接输出0或false;</div> <div>对 来说，如果左侧结果是true或者非零数；就可以直接输出1或者true;</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)

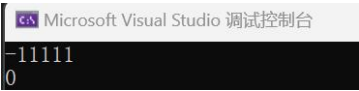
```
#include <iostream>
using namespace std;

int main()
{
    int pre=-11111, latter=11111;
    5 > 3 && 2 || pre == 8 < 4 - !0,
    latter = 0;
    cout << pre << endl << latter;
    return 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



8<4 - !0 这一部分没有进行计算，存在有短路运算；而***进行了计算，也就是不存在短路运算

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_所有_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

我=

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

int main()
{
    int i;

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

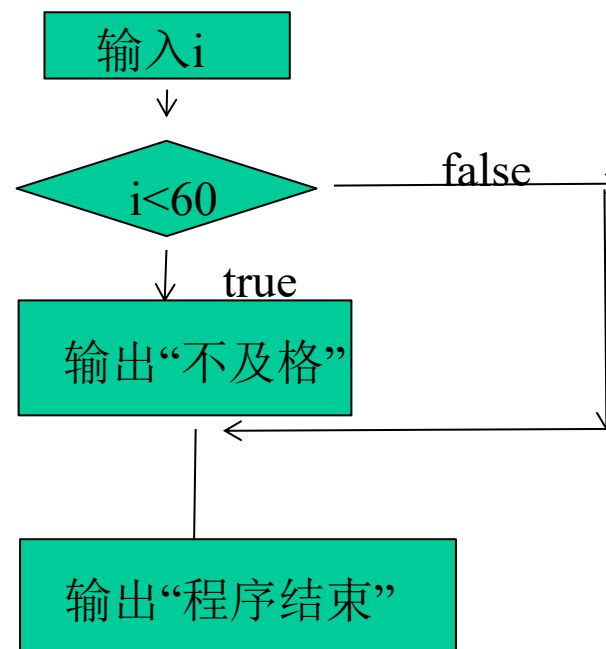
    if (i<60) {
        cout << "不及格" << endl;
    }
    cout << "程序结束" << endl;

    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





§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

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

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

int main()
{
    int i;

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

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

    return 0;
}
```

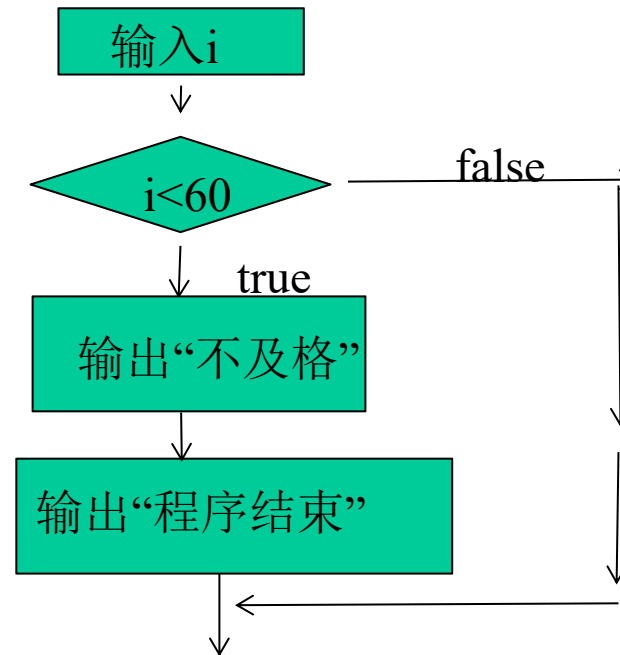
1、输入34，贴运行结果

```
Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

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

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



错误原因就是if里面加了分号，导致编译器认为这一句语法结束了。导致后面的大括号内容被判定为非法内容



§. 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

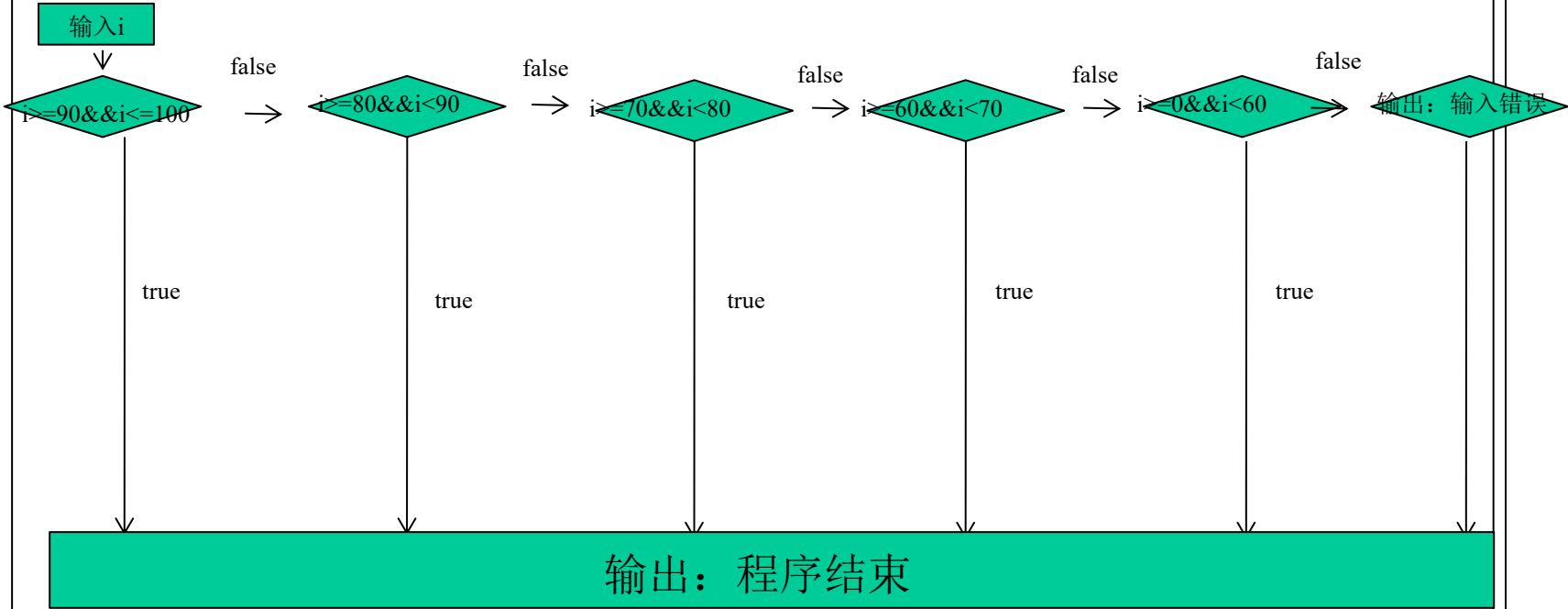
D. 观察下列程序的运行结果，回答问题

```
#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，但是i<90更好；和其他的语句一致，方便检查和阅读
- 3、i<90能否改为i<=90? 运行是否正确?
运行也可以正确



§. 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: }	<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>注意嵌套的语句的括号配对；每出现一个左括号都记录下来，直到出现右括号开始就与最后一个出现的左括号配对，以此类对</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____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，并给出该程序的程序及编译错误截图

2、请说明错误原因

是因为if下面必须配上一句else并且是顺序读取，如果编译器在读到if之后没出现else并且出现了一个其他语句就会报错

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main()  
5 {  
6     if (1==2) {  
7         if (2==3) {  
8             cout<<'a';  
9         }  
10        else {  
11            cout<<'b';  
12        }  
13        cout<<'c';  
14    }  
15    cout<<'f';  
16    else {  
17        if (3==4) {  
18            cout<<'d';  
19        }  
20        cout<<'e';  
21    }  
22 }  
23  
24
```

错误列表

代码	说明	项目	文件
E0127	应输入一个语句	cpp	源.cpp



§. 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__假__(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__任意__(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

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-->1-->3__

我不喜欢c哈哈哈；
感觉三目运算符像2这样用更加逻辑清晰一点



§. 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;

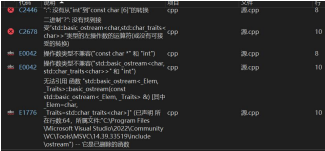
    a==1 ? "Hello" : 123; //编译报错

    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)

三目运算符在表达式中不能接受字符串或者输出的语句。只能接受字符常量，表达为整数，数字

§. 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)
101到109



-9到0





§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

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

    return 0;
}
```

在8.A的基础上

1、将6、8、default的位置进行了交换

2、将7写为常变量+常量形式

验证此程序与8.A的功能是否完全一致

(即：8.A中正确的, 此程序中同样正确；8.A错误的, 此程序中同样错误)

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
101
优

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
5
不及格

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
92
优

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
82
良

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
73
中

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
34
不及格

Microsoft Visual Studio 调试控制台
请输入成绩[0-100]
63
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)

```
#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后面不能跟变量，要跟常量；



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

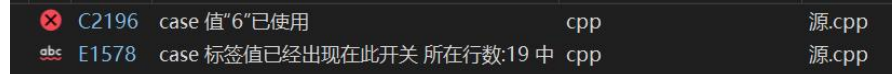
D. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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

重复的出现了case6这样一个情况；编译器顺序读取的时候就会出现错误



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

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

    return 0;
}
```

在8.A的基础上，将score从int改为float

1、给出编译错误的截图



2、解释错误原因


case后面不能接float型的常量;



§. 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>使得程序能跳出当下的循环，也就是不执行后续的判断；如果把该处break删除了，程序就会继续进行；</p>	
---	---	---



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

<pre>#include <iostream> using namespace std; int main() { int score; cout<<"请输入成绩[0-100]"<<endl; cin >> score; switch(score/10) { case 10: case 9: cout<<"优"<<endl; break; case 8: cout<<"良"<<endl; break; case 7: cout<<"中"<<endl; break; case 6: cout<<"及格"<<endl; break; case 5: case 4: case 3: case 2: case 1: case 0: cout<<"不及格"<<endl; break; default: cout<<"输入错误"<<endl; break; } return 0; }</pre>	<p>程序同8. A，将其改正确，即所有[0..100]之外的数据均给出“输入错误”即可</p> <pre>#include <iostream> using namespace std; int main() { int score; cout << "请输入成绩[0-100]" << endl; cin >> score; int score_1 = score; if (score_1 > 100 score_1 < 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; }</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：

[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

下一页

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）

下一页

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，而不能使用switch语句。因为switch有数据类型的限制，case后面不能接浮点型

4、总结switch语句使用时的注意事项

记得每一个case都要接break语句；case后面只能接常量，不能接变量；而且不能接浮点型数据

5、switch-case语句能完全取代if-else吗？

不能，很明显switch-case语句有数据类型的限制，而if-else没有。因此不能替代

需要文字描述的地方，尽可能用简单的文字描述清楚即可，不需要卷字数

§. 基础知识题 - 关系运算、逻辑运算与选择结构



```
int score;
cout << "请输入成绩[0-100]" << endl;
cin >> score;
switch (score*5 / 10) {
```

switch-case

关键我认为在于这个区间的取值，此时乘以5的话能让每个区间的左右边都除以10的时候有一个特殊的值，再根据case判断

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

if-else

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目