

§. 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、**10月10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 occupies most of the left side of the slide.

例：有效贴图

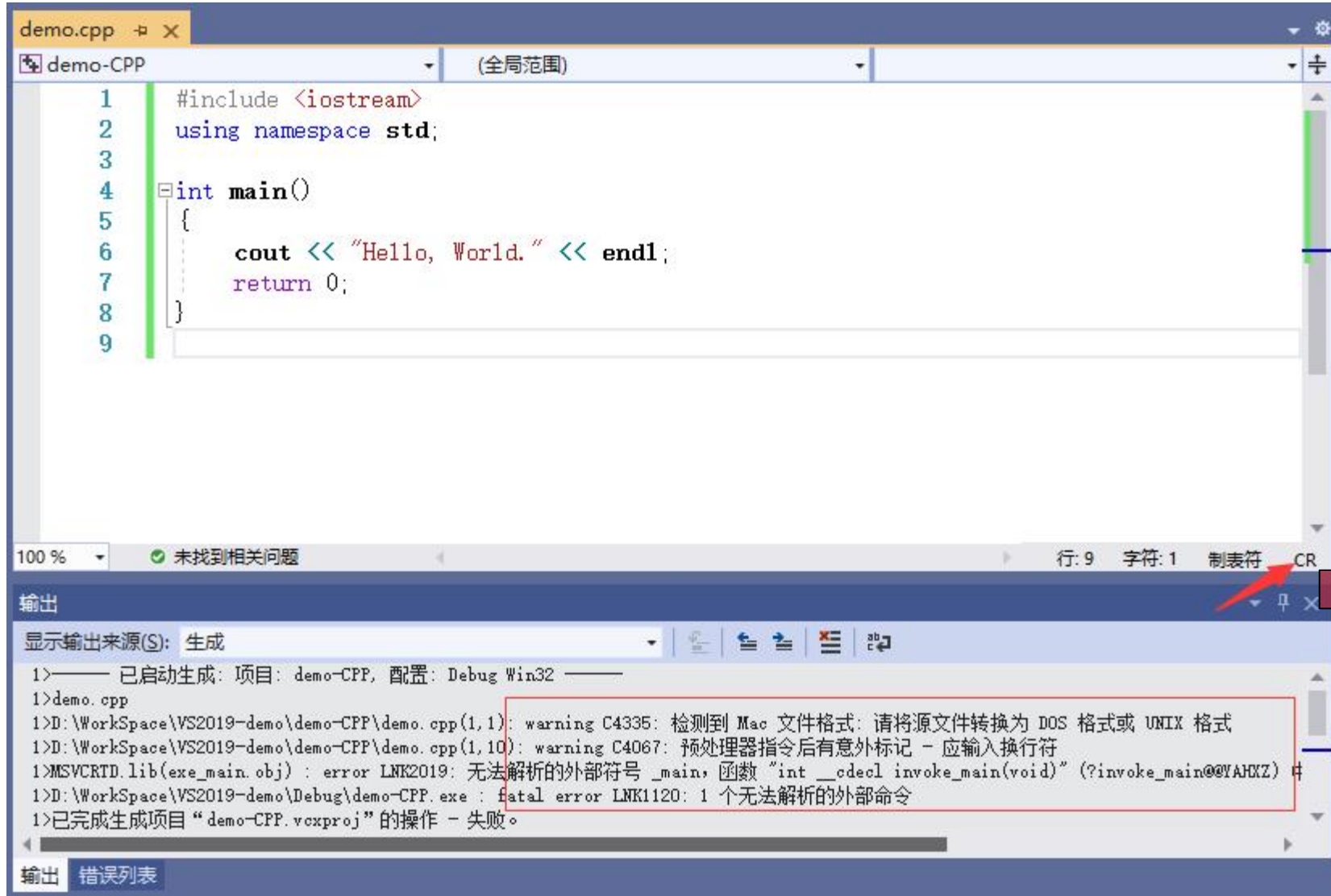
A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the "Hello, world!" line. The window is titled "Microsoft Visual Studio 调试控制台".



§. 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;

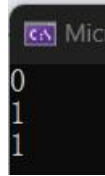
    d = a > b > c;
    cout << d << endl;

    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

```
cpp(8,11): warning C4804: ">": 在操作中使用类型 "bool" 不安全
cpp(11,11): warning C4804: "<": 在操作中使用类型 "bool" 不安全
cpp(14,11): warning C4804: "<": 在操作中使用类型 "bool" 不安全
```

报warning的原因:

连续比较的时候，会先计算第一组比较（以 $d=a>b>c$ 为例），因为 $a<b$ ，所以此时表达式 $a>b$ 的值赋为0，然后再去比较0和c的值。这与程序意图——比较三个变量值大小关系，是不同的，故会报warning。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

1、关系运算符的求值顺序

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

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

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

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

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

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

    return 0;
}
```

1、贴运行结果



- 2、 $a > b > c$ 这个式子，按人的常规理解($3 > 2$ 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.先比较 $a > b$ ，表达式为真，表达式的逻辑值为1，然后比较1和c的大小， $1 > c$ 不成立，故表达式的值为0，然后将表达式的值0赋给变量d，最终输出结果是0；
- 2.先比较 $a < b$ ，表达式为假，表达式的逻辑值为0，然后比较0和c的大小， $0 < c$ 成立，故表达式的值为1，然后将表达式的值1赋给变量d，最终输出结果是1；
- 3.先比较 $b > a$ ，表达式为假，表达式的逻辑值为0，然后比较0和c的大小， $0 < c$ 成立，故表达式的值为1，然后将表达式的值1赋给变量d，最终输出结果是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)

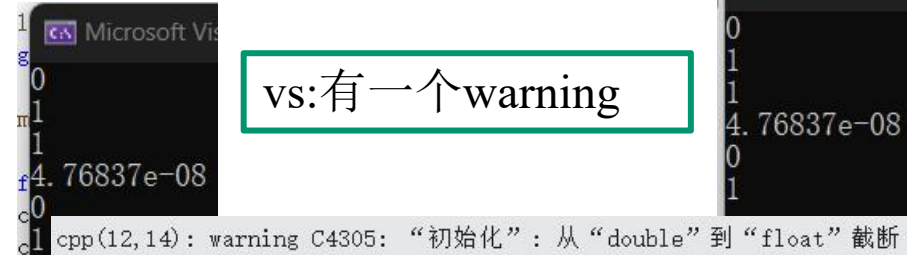
```
#include <iostream>
#include <cmath>
using namespace std;

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



vs:有一个warning

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



dev:

vs有warning, dev报错

3、由本例得出的结论，实数进行相等比较时的通用方法是 当两数相减的绝对值小于某个值则认为相等



§. 基础知识题 - 关系运算、逻辑运算与选择结构

2、关系运算符与实数

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

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

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

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

    return 0;
} //VS有两个warning
```

0
1
0
1
1
1

1、贴运行结果

```
(13,11): warning C4305: "初始化": 从 "double" 到 "float" 截断
(14,11): warning C4305: "初始化": 从 "double" 到 "float" 截断
```

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？

fabs (**) <1e-6相同，fabs (**) <1e-7不同。
原因：
1e-6=0.000001 1e-7=0.0000001而float型的精度在本题中是6位，故小数点后六位可以精确存储，第七位及之后的位数会有存储误差，再去相减的时候就会有误差，从而出现fabs (**) <1e-7与double型下不同的结果。（double型可以确保15~16位精度，七位左右是精确存储的）

§. 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

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)

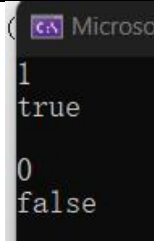
```
#include <iostream>
using namespace std;

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

    cout << endl;

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

    return 0;
}
```



```
1
true
0
false
```

1、贴运行结果

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

true在程序中被识别成布尔常量，输出时会输出其int型返回值 (true为1, false为0)

"true" 则是字符串类型，输出时会以字符串形式原样输出。

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)

```
std::cout << std::boolalpha;
```

```
bool value = true;
```

```
std::cout << value << std::endl; // 这样会打印 "true"
```

```
value = false;
```

```
std::cout << value << std::endl; // 这里会打印 "false"
```

并且验证得知，该前导格式控制符对后续bool值均有效



§. 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;

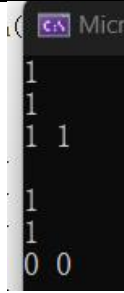
int main()
{
    bool k1 = true;

    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__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/变量在输出时，默认转换成int型的0/1输出



§. 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：



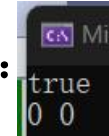
2、输入1，输出是：



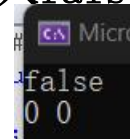
3、输入123，输出是：



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：

通过cin输入bool变量的值时，只能输入0或1，其他任何形式均为不合法，得到的均为不可信值。



§. 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;

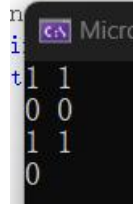
    k='A';
    cout << k << ' ' << (int)k << endl;

    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

报warning是由于等号两边的数据类型不同，赋值的时候会存在截断的情况，从而导致数据出错。

3、k='A' 是1字节赋值给1字节，为什么还有warning?

因为'A'的int值为其ASCII码65，而k为bool型，其值只有0/1，自然不可能是65，故会报错截断。

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=0000 0000 0000 0000 0000 0001 0000 0000

低位赋值，高位截断 1字节=0000 0000 k应该是0

实际是1 因为赋值时采用“非零为真零为假”的原则进行，任何非零数赋值给bool变量，结果均为1。

5、为什么不 cout << c, 而是 (int)c ?

因为c是char型，直接cout<<c, 会默认以char型输出，即输出ASCII码为0(截断后赋的值)的字符，可能不显示，故用int强制类型转换一下，可以得到c的值。

6、“非0为真0为假”这句话如何解释?

“非0为真0为假”适用于逻辑运算及赋值，对于逻辑变量来说，如果值不为0，返回的逻辑值就是1(true); 如果是0，则返回的逻辑值就是0(false)。

.cpp(7,9): warning C4305: “=”: 从“char”到“bool”截断
.cpp(13,9): warning C4305: “=”: 从“int”到“bool”截断
.cpp(16,12): warning C4305: “初始化”: 从“int”到“char”截断
.cpp(16,14): warning C4309: “初始化”: 截断常量值

§. 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 整型值 (0/1)

§. 基础知识题 - 关系运算、逻辑运算与选择结构



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

A. 完成下列两个表格的填写（a/b是两个逻辑值，填写的内容不要用黑色）

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

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



§. 基础知识题 - 关系运算、逻辑运算与选择结构

4、逻辑运算符与逻辑运算

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

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

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

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

    return 0;
}
```

1、贴运行结果

```
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. $a>b$ 表达式值为0

2. $m=0$

再由于逻辑与 (&&) 的短路特性，直接该整个表达式值为0，并且不会进行后半部分运算，故n的值没有变，仍未初始值1。

3、短路运算的意思是：无需执行下一个逻辑运算符就能求出解时，则下一个逻辑运算符不执行。



§. 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 m = 0;
    5 > 3 && 2 || 8 < 4 - (m = !0);
    cout << m << endl;
    cout << (5 > 3 && 2 || 8 < 4 - !0,
1 && 0) << endl;
    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

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;
}
```

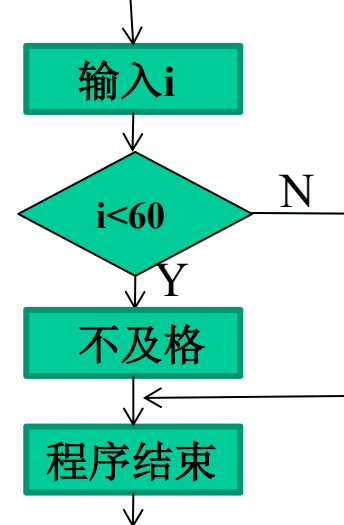
1、输入34，贴运行结果

```
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;
}
```

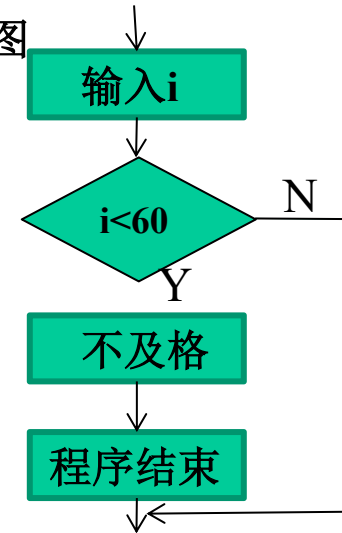
1、输入34，贴运行结果

```
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
```

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

```
cpp(11,11): warning C4552: " < ": 未使用表达式结果
cpp(11,15): error C2429: 语言功能 "if/switch 中的 init-statement" 需要编译器标志 "/std:c++17"
cpp(11,16): error C2059: 语法错误: ")"
cpp(11,18): error C2143: 语法错误: 缺少 ";" (在 "{" 的前面)
```

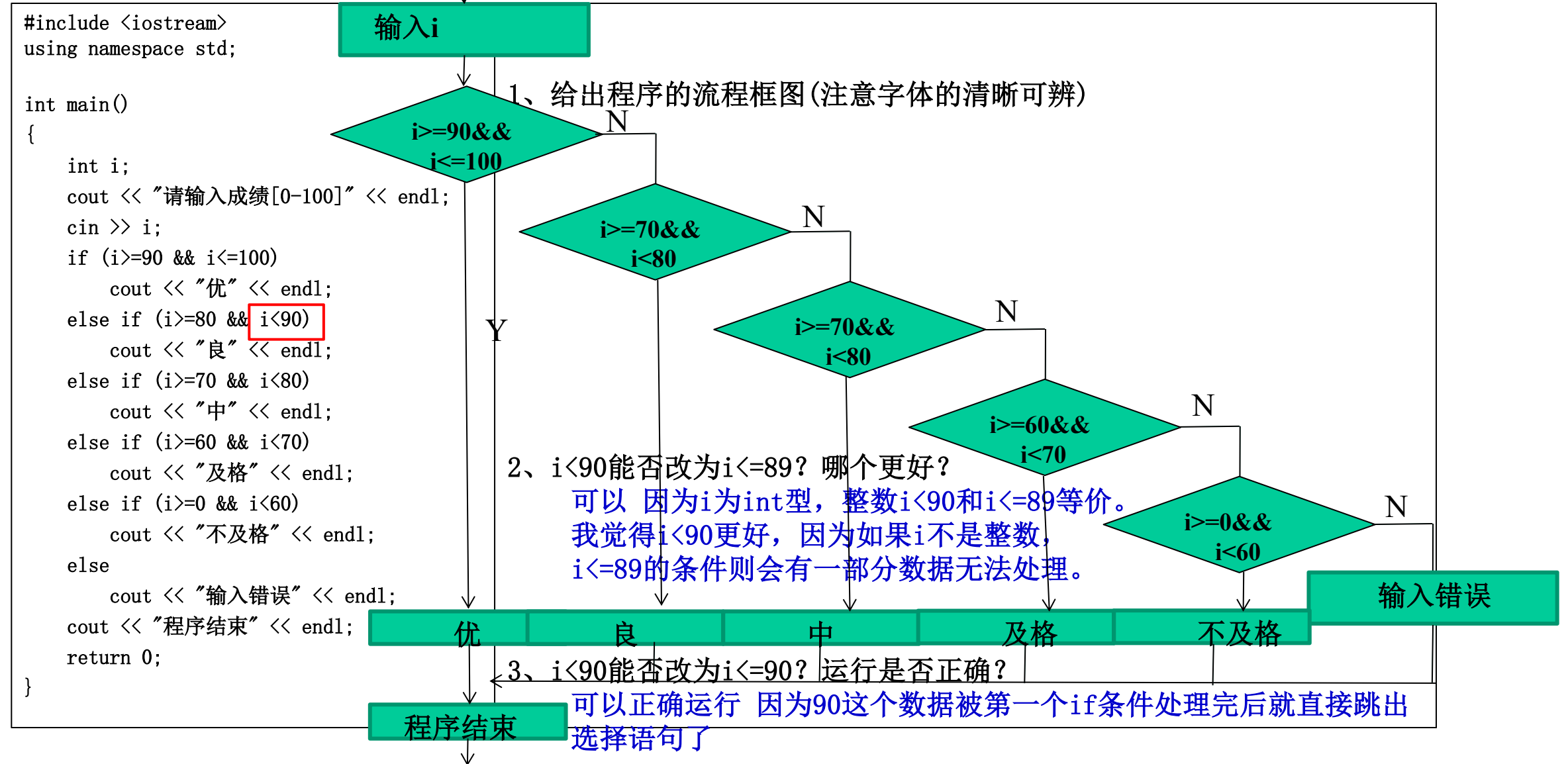
if语句运用时，判断条件的表达式没有“;”
报错的所有原因都是因为“;”的不恰当使用。
“;”会使得 if 语句的条件被忽略，而随后的
代码块将始终执行，不受 if 条件的影响。



§. 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



§. 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

利用栈理解：遇{进栈，遇}则栈顶{出栈并匹配为一对

§. 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，并给出该程序的程序及编译错误截图

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main()  
5  {  
6      int i;  
7      cin >> i;  
8      if (i>2) {  
9          if (i<100) {  
10             i=i+1;  
11         }  
12         else {  
13             i=i*2;  
14         }  
15         i=i/8;  
16     }  
17     i=i;  
18     else {  
19         if (i>0) {  
20             i=15+i;  
21         }  
22         i=15-i;  
23     }  
24     cout << i << endl;
```

cpp(18, 5): error C2181: 没有匹配 if 的非法 else

2、请说明错误原因

整个多分支语句可以看作一整个语句，中间不允许插入任何其他语句。



§. 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

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（最喜欢）、//3、//1（最不喜欢）



§. 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>1、给出编译报错的截图</div> <div></div> <div>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)</div> <div><div>表达式2和3的类型必须相容。 第一句表达式2类型为字符串类型，表达式3为int型，不相容，故报错； 第二句表达式2类型为cpp下的输出格式，表达式3为c下的输出格式，不相容，故报错； 第三句表达式2为字符型，表达式3为int型，可以相容，所以不报错。</div></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)

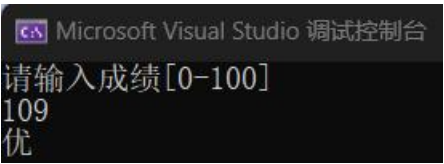
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;
            break;
        case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

(100,110) 和 (-10,0) 区间内的数据会被错误处理。这些都应该落入default运行，但结果是:





§. 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> <ul 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：8. A和8. B <u>完全一致</u> (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

```
\Users\HP\Desktop\sj\Project1\104.cpp(20,12): error C2131: 表达式的计算结果不是常数
C:\Users\HP\Desktop\sj\Project1\104.cpp(20,10):
因读取超过生命周期的变量而失败
C:\Users\HP\Desktop\sj\Project1\104.cpp(20,10):
请参见“k”的用法
\Users\HP\Desktop\sj\Project1\104.cpp(20,12): error C2051: case 表达式不是常量
```

2、解释错误原因

k变成了int型变量，并且在输入的时候并未输入k的值，导致k的值无法确定，此时表达式的值也无法确定，故报错。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

C:\Users\HP\Desktop\sj\Project1\104.cpp(20,12): error C2196: case 值“6”已使用

2、解释错误原因

case 4+2与case 6的值重复使用了，故报错。（case的标签值不能重复使用）



§. 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

```
C:\Users\HP\Desktop\sj\Project1\104.cpp(8,19): error C2450: 类型为“float”的 switch 表达式无效
C:\Users\HP\Desktop\sj\Project1\104.cpp(8,19):
    要求整型表达式
```

2、解释错误原因

switch-case语句后跟的表达式最后一定要转换成整型表达式才可以，不然就会报错。



§. 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



以及其他在[80,90) 区间内的数据

2、解释break的作用

break语句的作用是在执行完对应的case语句后就跳出选择结构。如果没有break，则会按照顺序继续向下执行其他的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)

<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> <div>仅考虑输入正确的情况（即输入的成绩均为整数）</div>	<pre>#include <iostream> using namespace std; int main() { int score; cout<<"请输入成绩[0-100]"<<endl; cin >> score; if (score<=100&&score>=0) { switch(score/10) { case 10: case 9: cout<<"优"<<endl; break; case 8: cout<<"良"<<endl; break; case 7: cout<<"中"<<endl; break; case 6: cout<<"及格"<<endl; break; case 5: case 4: case 3: case 2: case 1: case 0: cout<<"不及格"<<endl; break; } } else{ cout<<"输入错误"<<endl; } return 0; }</pre>
---	--	--



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

H. 思考

如果将成绩区间对应为：[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

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

int main()
{
    int i;
    cout << "请输入成绩[0-100]" << endl;
    cin >> i;
    if (i >= 84 && i <= 100)
        cout << "优" << endl;
    else if (i >= 68 && i < 84)
        cout << "良" << endl;
    else if (i >= 55 && i < 68)
        cout << "及格" << endl;
    else if (i >= 0 && i < 55)
        cout << "不及格" << endl;
    else
        cout << "输入错误" << 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）

```
switch (score / 10) {
    case 10:
    case 9:
        cout << "优" << endl;
        break;
    case 7:
    case 6:
    case 5:
    case 4:
    case 3:
    case 2:
    case 1:
    case 0:
        cout << "不及格" << endl;
        break;
}
```

```
switch (score) {
    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 69:
    case 68:
        cout << "良" << endl;
        break;
    // ... (other cases)
}
```

首先根据成绩区间可以将一部分/10后处理，还有一部分则需要一分一分地处理。

3、如果学生成绩带小数点，即“xx.5”形式，能用if语句吗？能用switch语句吗？请解释原因
能用if语句，不能用switch语句。因为switch语句处理的都是整型表达式，不能处理小数。



H续

4、总结switch语句使用时的注意事项

1. switch语句后必须跟整型表达式;
2. case后跟的整型常量表达式应该互不相同, 但顺序无所谓;
3. 各语句序列最后一句应该是break, 否则连续执行下一个case;
4. 若数据精确到小数点后, 则无法使用switch。

5、switch-case语句能完全取代if-else吗?

不能。因为根据前面论述可知, 一旦要处理的数据为小数, switch就不能用了, 因为其后必须跟整型表达式。

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



此页不要删除，也没有意义，仅仅为了分隔题目