

§. 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

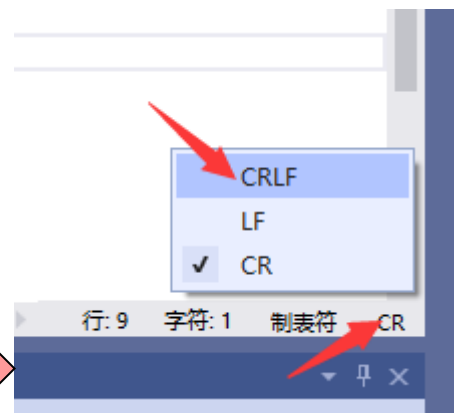
The screenshot shows the Visual Studio 2022 IDE with a C++ file named `demo.cpp`. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World." << endl;
7     return 0;
8 }
9
```

The status bar at the bottom of the editor indicates "行: 9 字符: 1 制表符: CR". Below the editor, the "输出" (Output) window shows the following messages:

```
1>—— 已启动生成: 项目: demo-CPP, 配置: Debug Win32 ——
1>demo.cpp
1>D:\WorkSpace\VS2019-demo\demo-CPP\demo.cpp(1,1): warning C4335: 检测到 Mac 文件格式: 请将源文件转换为 DOS 格式或 UNIX 格式
1>D:\WorkSpace\VS2019-demo\demo-CPP\demo.cpp(1,10): warning C4067: 预处理器指令后有意外标记 - 应输入换行符
1>MSVCR7D.lib(exe_main.obj) : error LNK2019: 无法解析的外部符号 _main, 函数 "int __cdecl invoke_main(void)" (?invoke_main@YAHXZ) 中
1>D:\WorkSpace\VS2019-demo\Debug\demo-CPP.exe : fatal error LNK1120: 1 个无法解析的外部命令
1>已完成生成项目 "demo-CPP.vcxproj" 的操作 - 失败。
```

The error messages are highlighted with a red box. The status bar at the bottom of the Output window shows "行: 9 字符: 1 制表符: CR".



§. 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;

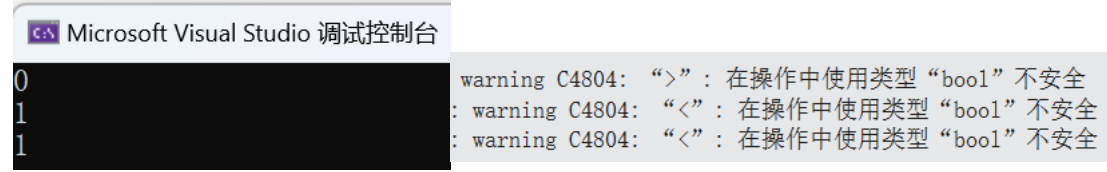
    d = a > b > c;
    cout << d << endl;

    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

对d赋值的语句进行运算的时候，按照顺序，先进行第一个大小比较，得到一个bool值0/1，bool值和int比较得到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;

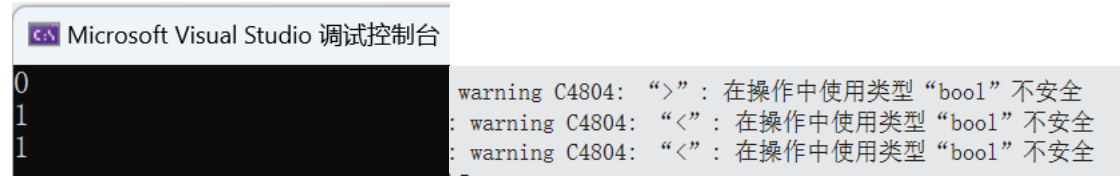
    d = a > b > c;
    cout << d << endl;

    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)

$3 > 2$ 的结果是真，故值为1，而 $1 > 1$ 结果是假，故最终值为0。
 $3 < 2$ 的结果是假，故值为0，而 $0 < 1$ 结果是真，故最终值为1。
 $2 > 3$ 的结果是假，故值为0，而 $0 < 1$ 结果是真，故最终值为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)

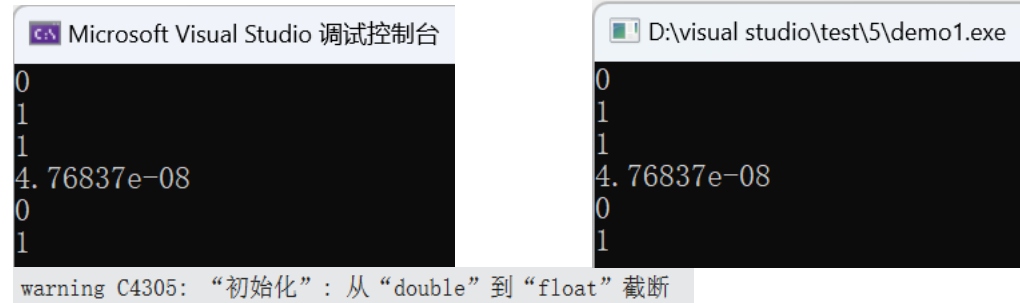
```
#include <iostream>
#include <cmath>
using namespace std;

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 当两数相减的绝对值小于某个给定允许误差值则认为相等



§. 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)

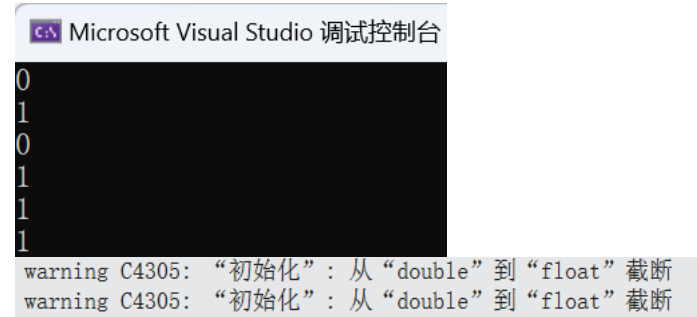
```
#include <iostream>
#include <cmath> //VS可不加
using namespace std;

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
```

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？

double 不同，float 相同。

float 型的精度只有 7 位，当误差设置达到 $1e-6$ ， $1e-7$ 时，已经超过了 float 型的有效数字位数，无法准确比较，因此相同且为真；但是 double 为 15 位有效数字，在该误差设置范围内能够有效反映是否相等，因此不同。



§. 基础知识题 – 关系运算、逻辑运算与选择结构

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

3、逻辑常量与逻辑变量

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

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

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

    cout << endl;

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

    return 0;
}
```

1、贴运行结果

```
Microsoft Visual Studio 调试控制台
1
true
0
false
```

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

true/false是逻辑常量，输出是1/0，"true/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)

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      bool t = true;
7
8      cout << t << endl;
9      cout << boolalpha << t << endl;
10     return 0;
11 }
```

```
Microsoft Visual Studio 调试控制台
1
true
```



§. 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;

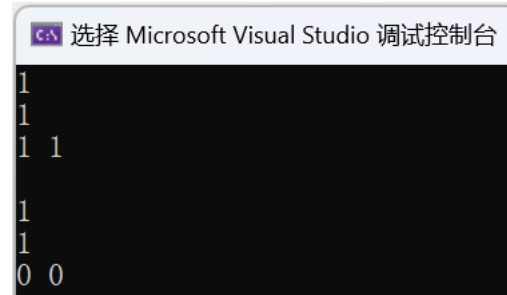
    cout << sizeof(true) << endl;
    cout << sizeof(k1) << endl;
    cout << k1 << ' ' << int(k1) << endl;

    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__1/0__

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)

bool型数据在输出只输出0/1数值，true输出1，false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)

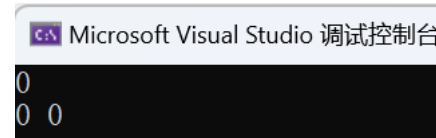
```
#include <iostream>
using namespace std;

int main()
{
    bool k;

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

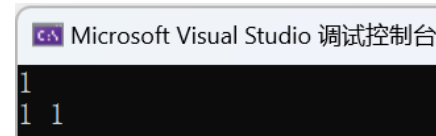
    return 0;
}
```

1、输入0，输出是：



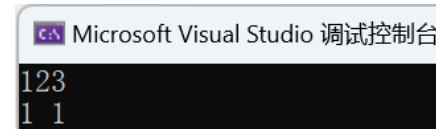
Microsoft Visual Studio 调试控制台
0
0 0

2、输入1，输出是：



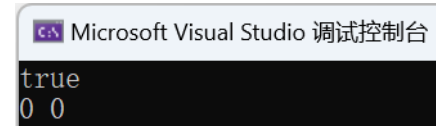
Microsoft Visual Studio 调试控制台
1
1 1

3、输入123，输出是：



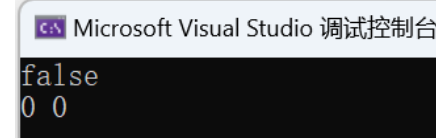
Microsoft Visual Studio 调试控制台
123
1 1

4、输入true，输出是：



Microsoft Visual Studio 调试控制台
true
0 0

5、输入false，输出是：



Microsoft Visual Studio 调试控制台
false
0 0

总结bool型变量在输入时的规则：

在输入数值时，
输入非0（或真值）
为真，bool值为1，
输入为0（或假值）
为假，bool值为0；
在输入字符串时，
无论输入什么都
为假，bool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;

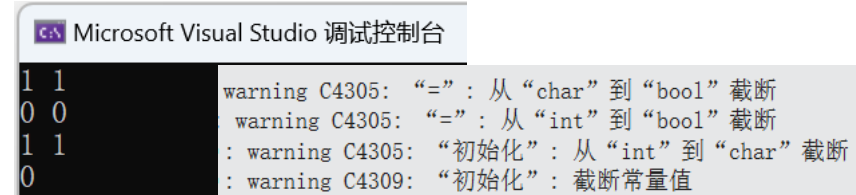
    k=0;
    cout << k << ' ' << (int)k << endl;

    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

char, int型数据赋值给bool型数据，int型数据赋值给char型数据都会发生高位截断，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?

bool型的值只能是0/1，机内存储为0000/0001，用1字节char型给bool型赋值还是会产生截断，造成数据丢失。

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的机内存储末尾为0000，若按此赋值给bool型，则bool型应为0，但是bool型赋值遵循非0为真，0为假，因此实际上k为1。

5、为什么不 cout << c，而是 (int)c ？

cout << c 输出的是一个char型数据，是256赋值给char型c后ASCII码所对应的字符，使用(int)进行强制类型转换，输出c的ASCII码值，才是int型数值。

6、“非0为真0为假”这句话如何解释？

非0的数值（或真值）在给bool型赋值时，bool值为1，即true；0（或假值）给bool型赋值时，bool值为0，即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

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)

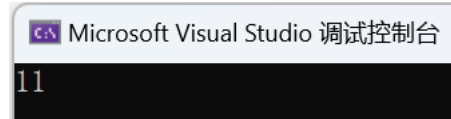
```
#include <iostream>
using namespace std;

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 1/0参与运算



§. 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)

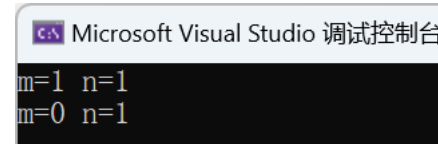
```
#include <iostream>
using namespace std;

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



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

- ① $a>b$, 为假
- ② $m=a>b$, 得到 $m=0$, 后续不再执行
- ③ $(m=a>b)\&\&(n=c>d)$, 整个表达式值为0

3、短路运算的意思是:

当逻辑运算符为 $\&\&$ 时, $\&\&$ 前的表达式结果为假, bool型值为0, 那么 $\&\&$ 后的表达式被短路, 对整个表达式结果没有影响, 整个表达式结果为假, 值为0;

当逻辑运算符为 $\|\|$ 时, $\|\|$ 前的表达式结果为真, bool型值为1, 那么 $\|\|$ 后的表达式被短路, 对整个表达式结果没有影响, 整个表达式结果为真, 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;
}
```

Microsoft Visual Studio 调试控制台

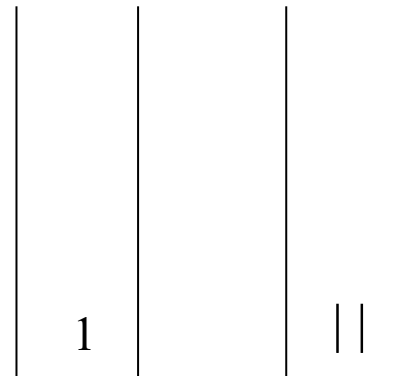
```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a, x, b, y;
6      a = x = 100;
7      b = y = 10;
8      cout << a << ' ' << b << endl;
9      (a = 5 > 3 && 2) || (b = 8 < 4 - !0);
10     cout << a << ' ' << b << endl;
11     cout << (5 > 3 && 2 || 8 < 4 - !0) << endl;
12     cout << x << ' ' << y << endl;
13     (x = 5 < 3 && 5 < 2) || (y = 8 > 4 - !0);
14     cout << x << ' ' << y << endl;
15     cout << (5 < 3 && 5 < 2 || 8 > 4 - !0) << endl;
16     return 0;
17 }
```

```
100 10
1 10
1
100 10
0 1
1
```

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, 5<3 && 5<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、8>4 - !0 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）



到!为止全部忽略



§. 基础知识题 – 关系运算、逻辑运算与选择结构

此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 关系运算、逻辑运算与选择结构

5、if语句 - 基本使用

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

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

int main()
{
    int i;

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

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

    return 0;
}
```

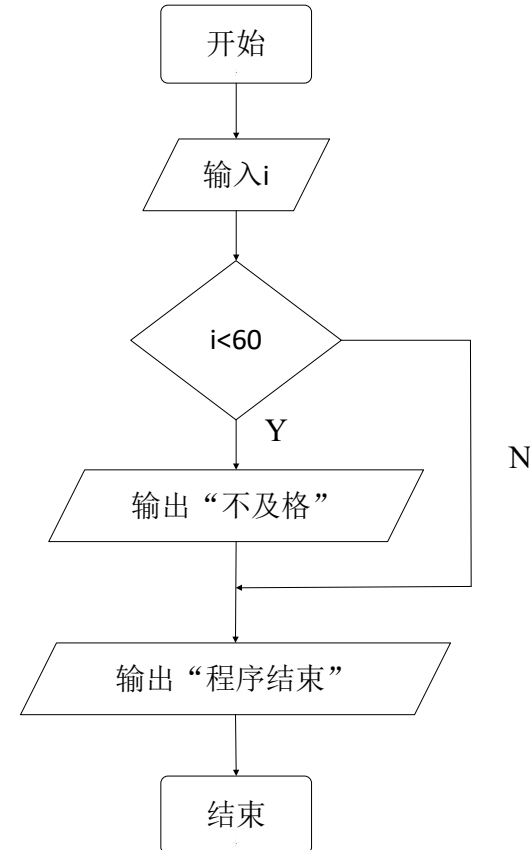
1、输入34，贴运行结果

2、输入74，贴运行结果

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

```
Microsoft Visual Studio 调试控制台
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;
}
```

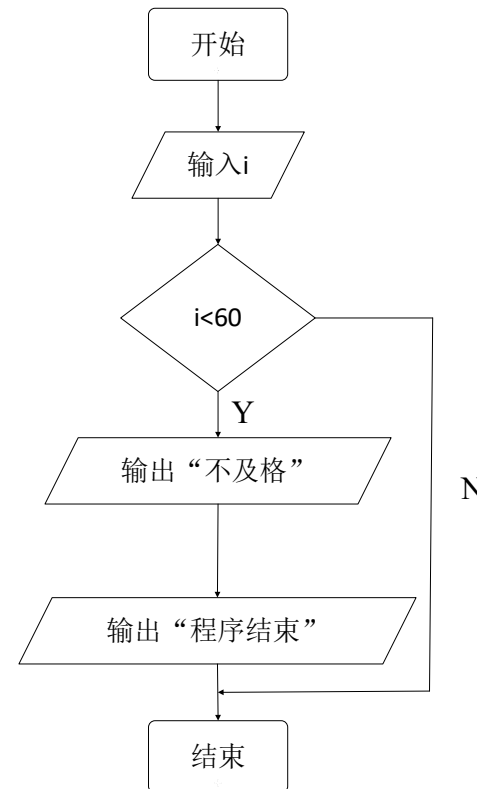
1、输入34，贴运行结果

2、输入74，贴运行结果

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

```
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

```
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)
```

- ①计算机读取时，读取到<时，后面识别到是一个“60; ”而不是60，故格式错误，所以警告未使用表达式结果。
- ②; 表示语句结束，（后面没有“; ”报错
- ③;表示语句结束，所以提示在; 前缺少一个)
- ④if括号内只能是表达式，而且if () 后面不能加;



§. 基础知识题 - 关系运算、逻辑运算与选择结构

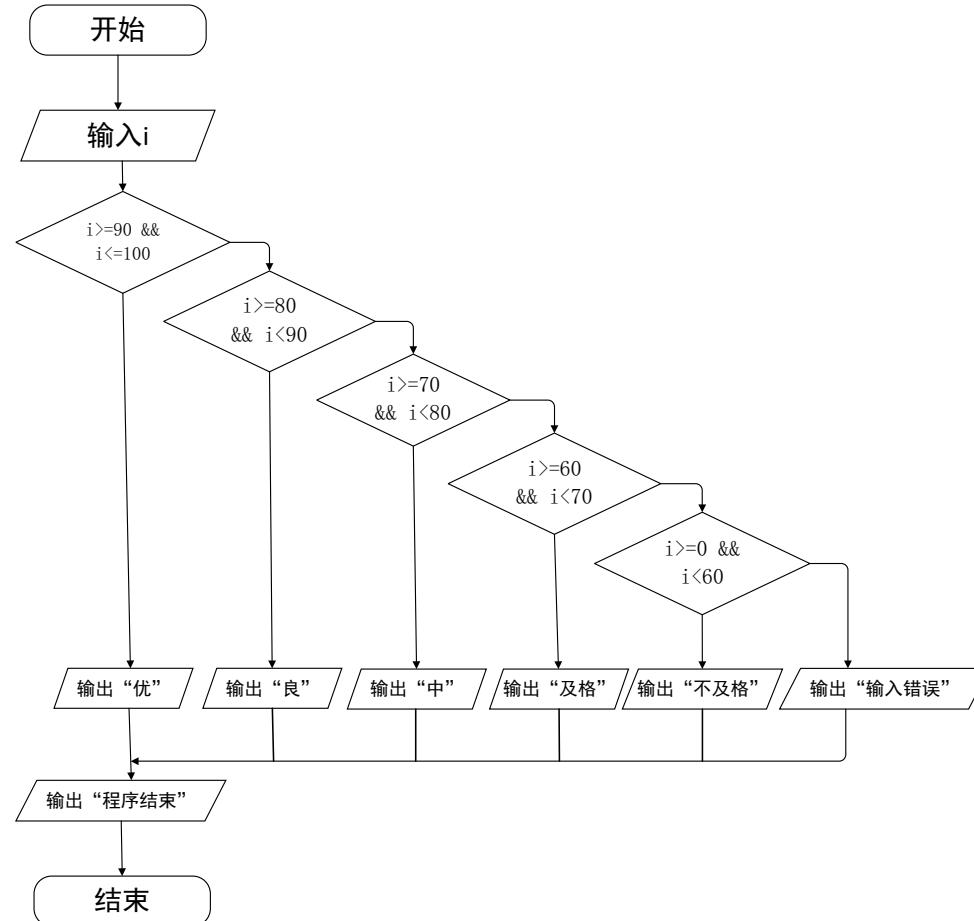
5、if语句 - 基本使用

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? **i此时为int型，故可以，但是i<90更好。**
- 3、i<90能否改为i<=90? 运行是否正确? **能，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 } 和在其上面离其最近的 { 进行配对
---	---



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套

B. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    B;  
}
```

- 1、当表达式1__真__(真/假/任意)，表达式2__真__(真/假/任意)时，
执行语句A
- 2、当表达式1__真__(真/假/任意)，表达式2__任意__(真/假/任意)
时，
执行语句B



§. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套

C. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    else {  
        B;  
    }  
    C;  
}  
else {  
    if (表达式3) {  
        D;  
    }  
    E;  
}
```

- 1、当表达式1__真__(真/假/任意)，表达式2__真__(真/假/任意)时，执行语句A
- 2、当表达式1__真__(真/假/任意)，表达式2__假__(真/假/任意)时，执行语句B
- 3、当表达式1__真__(真/假/任意)，表达式2__任意__(真/假/任意)时，执行语句C
- 4、当表达式1__假__(真/假/任意)，表达式3__真__(真/假/任意)时，执行语句D
- 5、当表达式1__假__(真/假/任意)，表达式3__任意__(真/假/任意)时，执行语句E



§. 基础知识题 – 关系运算、逻辑运算与选择结构

6、if语句 – 多重嵌套

D. 一个if语句嵌套如下，回答问题

```
if (表达式1) {  
    if (表达式2) {  
        A;  
    }  
    else {  
        B;  
    }  
    C;  
}  
→ F;  
else {  
    if (表达式3) {  
        D;  
    }  
    E;  
}
```

在6.C的基础上，在箭头位置插入语句F

1、请构造一个符合此要求的测试程序，并给出该程序的程序及编译错误截图

```
1  #include <iostream>  
2  using namespace std;  
3  int main()  
4  {  
5      if (3>2) {  
6          if (2>1) {  
7              cout << "a";  
8          }  
9          else {  
10             cout << "b";  
11         }  
12         cout << "c";  
13     }  
14     cout << "false";  
15     else {  
16         if (4>3) {  
17             cout << "d";  
18         }  
19         cout << "e";  
20     }  
21     return 0;  
22 }
```

error C2181: 没有匹配 if 的非法 else

2、请说明错误原因

当计算机读完if内的语句后，读cout语句；意味着if语句结束，因此后续的else部分被认为是单独部分，计算机没有找到与之匹配的if，故报错。



§. 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>	<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 Visual Studio 调试控制台
12 34
max=34
max=34
max=34
max=34
```

2、输入34 12，给出运行截图

```
Microsoft Visual Studio 调试控制台
34 12
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 //3 //1



§. 基础知识题 - 关系运算、逻辑运算与选择结构

7、条件运算符与条件表达式

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

<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>error C2446: “:” : 没有从 “int” 到 “const char [6]” 的转换</div> <div>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)</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>条件表达式中，表达式2和表达式3的类型必须相容。</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

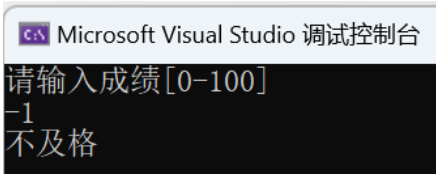
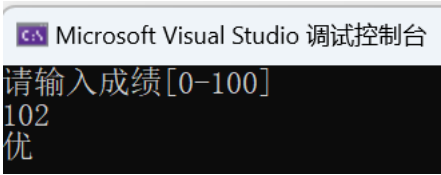
A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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)





§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

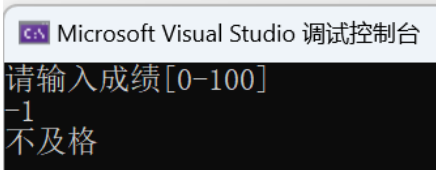
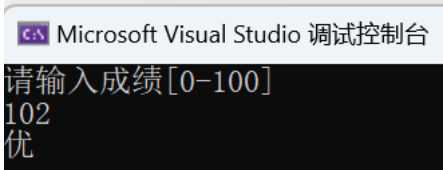
B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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)



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> <p>1、给出编译错误的截图</p> <div><p>error C2131: 表达式的计算结果不是常数</p><p>error C2051: case 表达式不是常量</p></div> <p>2、解释错误原因</p> <p>case后必须是整型常量表达式，但是int k是变量，表达式不符合格式要求。</p>
--	--



§. 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

在case 4+2语句前，已经有case 6，两个表达式重复，case后的各个整型常量表达式的值应该各不相同。



§. 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> <div>error C2450: 类型为“float”的 switch 表达式无效</div> <p>2、解释错误原因</p> <p>switch（）内和case后表达式最后取值一定要是整数，而score为float型，score/10最后结果也为float型，结果不一定为整数。</p>
---	--



§. 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)

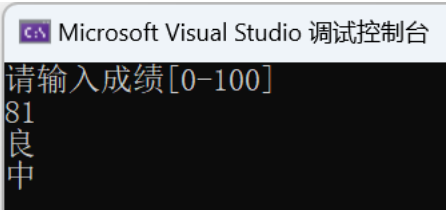
```
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"请输入成绩[0-100]"<<endl;
    cin >> score;
    switch(score/10) {
        case 10:
        case 9:
            cout<<"优"<<endl;
            break;
        case 8:
            cout<<"良"<<endl;

        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

跳出switch结构，结束case语句下的程序执行。



§. 基础知识题 - 关系运算、逻辑运算与选择结构

8、switch-case语句

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

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

    return 0;
}
```

程序同8. A，将其改正确，即所有[0..100]之外的数据均给出“输入错误”即可

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int score;
6      cout << "请输入成绩[0-100]" << endl;
7      cin >> score;
8      if (score >= 0 && score <= 100) {
9          switch (score / 10) {
10             case 10:
11             case 9:
12                 cout << "优" << endl;
13                 break;
14             case 8:
15                 cout << "良" << endl;
16                 break;
17             case 7:
18                 cout << "中" << endl;
19                 break;
20             case 6:
21                 cout << "及格" << endl;
22                 break;
23             case 5:
24             case 4:
25             case 3:
26             case 2:
27             case 1:
28             case 0:
29                 cout << "不及格" << endl;
30                 break;
31             }
32         }
33         else
34             cout << "输入错误" << endl;
35
36         return 0;
37     }
```



§. 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

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)

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int score;
6      cout << "请输入成绩[0-100]" << endl;
7      cin >> score;
8      if (score >= 0 && score <= 100) {
9          if (score >= 0 && score < 70) {
10             if (score >= 0 && score < 60)
11                 cout << "不及格" << endl;
12             else
13                 cout << "及格" << endl;
14         }
15         else {
16             if (score >= 70 && score < 85)
17                 cout << "良" << endl;
18             else
19                 cout << "优" << endl;
20         }
21     }
22     else
23         cout << "输入错误" << endl;
24
25     return 0;
26 }
```

```
3  int main()
4  {
5      int score;
6      cin >> score;
7      if (score >= 0 && score <= 100) {
8          switch (score) {
9              case 100:
10             case 99:
11             case 98:
12             case 97:
13             case 96:
14             case 95:
15             case 94:
16             case 93:
17             case 92:
18             case 91:
19             case 90:
20             case 89:
21             case 88:
22             case 87:
23             case 86:
24             case 85:
25                 cout << "优" << endl;
26                 break;
27             }
```

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，不能用switch语句，因为switch只能精确到整数位。

4、总结switch语句使用时的注意事项

switch括号内都为整形表达式，case后为整型常量表达式且不能重复，switch可能在数据范围两侧有溢出现象，执行完后break可以结束case执行。

5、switch-case语句能完全取代if-else吗？

不能

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