

# §. 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 title is "Microsoft Visual Studio 调试控制台". The output text is:

```
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. . .
```

The screenshot is a full-width capture of the console window, which is considered an invalid example according to the requirements.

例：有效贴图

A screenshot of the Microsoft Visual Studio debug console window, but it is cropped to show only the first line of output. The window title is "Microsoft Visual Studio 调试控制台". The output text is:

```
Hello, world!
```

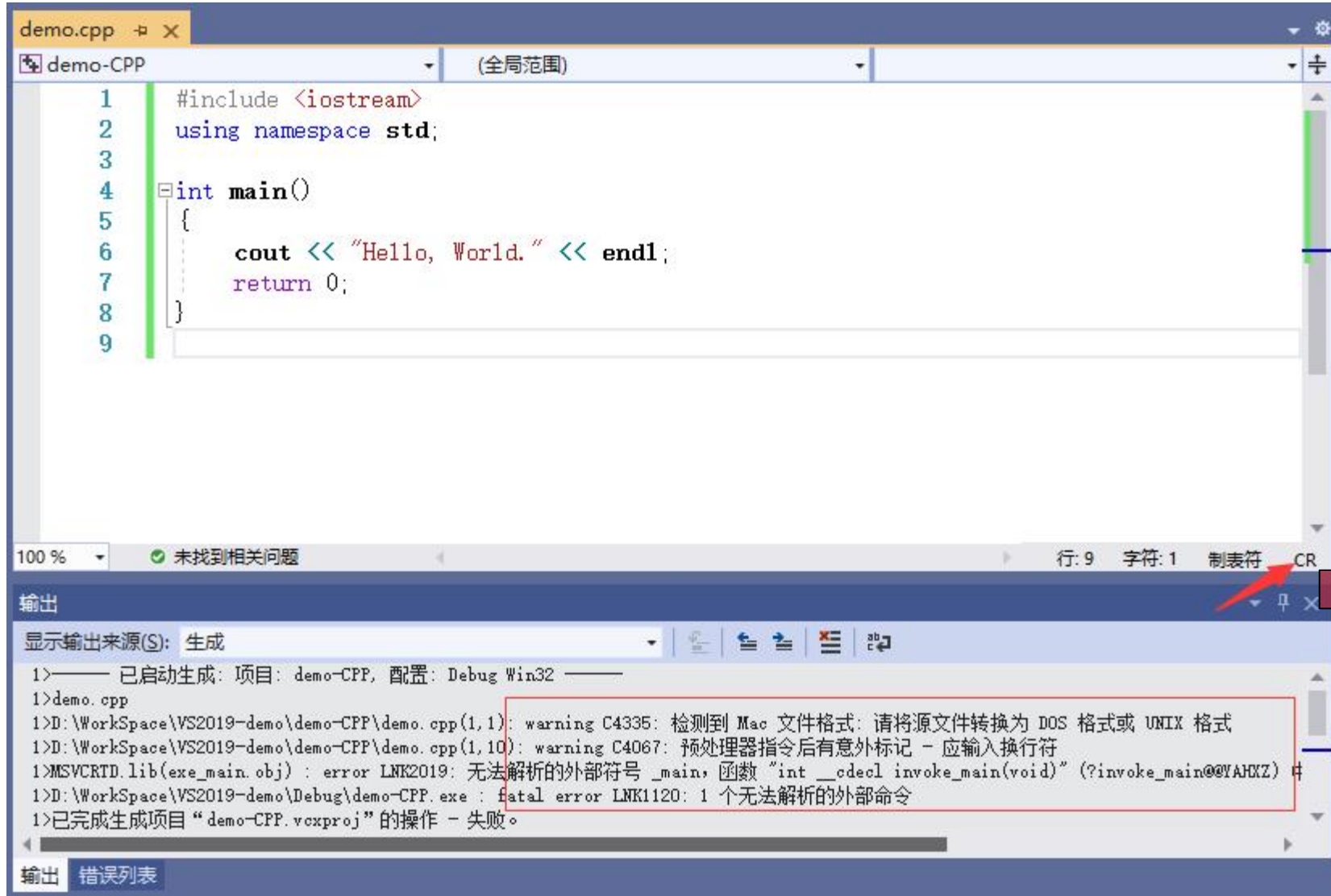
This is considered a valid example because it only captures the effective part of the output.



## §. 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

比较时不能连续比较，比较结果会输出1或0而并非变量本身的值，因此在比较时不应使用连续的比较。





## §. 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)

计算机从左向右扫描首先遇见a、b比较为真，输出1与c比较则输出0；同理其余也是相同的原因。计算机比较后输出的结果为真或假而并非变量的值。

## §. 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 引用cmath头文件并看计算机有无浮点数误差再用==比较

Microsoft Vis	D:\Document
0	0
1	1
1	1
4.76837e-08	4.76837e-08
0	0
1	1

Microsoft Vis
0
1
1
4.76837e-08
0
1

信息  
In function 'int main()':  
[Error] 'fabs' was not declared in this scope; did you mean 'labs'?





## §. 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;

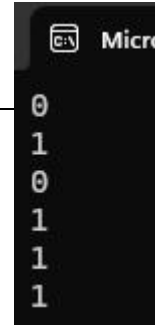
int main()
{
    double d1=123.456789012345678;
    double d2=123.456789123456789;
    cout << (d1==d2) << endl;
    cout << (fabs(d1-d2)<1e-6) << endl;
    cout << (fabs(d1-d2)<1e-7) << endl;

    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？  
在 float 下相同，在 double 型下不同，由于 double 的精度高于 float，因此在精度上 double 会显示出不同的结果，d1 与 d2 也不认为是相等的。



0  
1  
0  
1  
1  
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

### 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

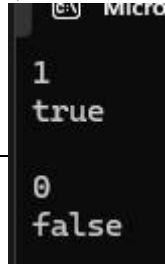
2、解释 true 和 "true" 的区别 (false和"false")

true是判断符，输出1； "ture" 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)  
在cout时使用boolalpha使true和false输出位单词。



```
1
true

0
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)  
非0为真0为假

1  
1  
1 1  
  
1  
1  
0 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：当输入结果是数字时0代表假，若数字超过一位则会截断输出第一位的数字；遇见非数字则会输入0，即假



## §. 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

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

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

2、解释VS下warning的意思

cout后面直接加k，由于k是bool型因此只能输出第一位数字所以会发生截断。

3、k='A' 是1字节赋值给1字节，为什么还有warning?

A的值是65因此输出为1，会报warning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)

按4字节赋值给1字节，k应该是0; 实际是1，因为2非零，所以输出为1。

5、为什么不 cout << c，而是 (int)c ?

c为char型，需要强制转换为int型再输出

6、“非0为真0为假”这句话如何解释?

bool赋值时不是多字节赋少字节的规则，因此低bit为都为0也可能是1，所以非0的数字为真。

## §. 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0或1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	假	假	真	真
1	0	假	真	假	真
0	1	真	假	假	真
0	0	真	真	假	假

a	b	!a	!b	a&&b	a  b
非0	非0	假	假	真	真
非0	0	假	真	假	真
0	非0	真	假	假	真
0	0	真	真	假	假

## §. 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)

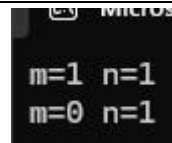
```
#include <iostream>
using namespace std;

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 $a>b$  (0) ② $m=0$  ③直接计算 $\&\&$ 为0

3、短路运算的意思是：C++中的运算符短路指的是在逻辑运算中, 当结果可以确定时, 后续的计算就被忽略了。



## §. 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()
{
    int a = 1;
    5 > 3 && 2 || (a = 8 < 4 - !0);
    cout << int(8 < 4 - !0) << endl;
    cout << a << 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）

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

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

    return 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

### 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

3、画出程序对应的流程框图

请输入成绩[0-100]

34

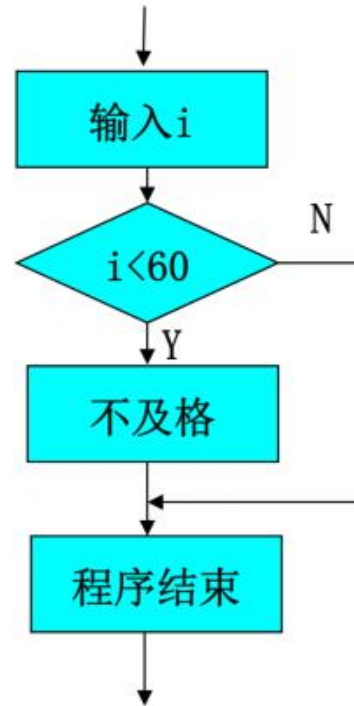
不及格

程序结束

请输入成绩[0-100]

74

程序结束





## §. 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] 请输入成绩[0-100]

34

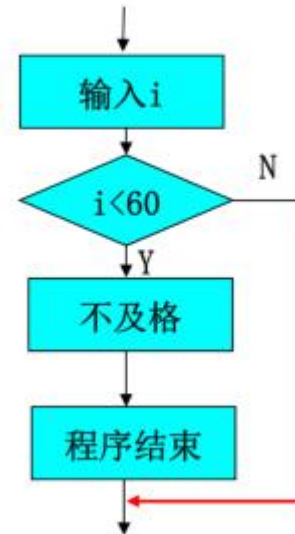
74

2、输入74，贴运行结果

不及格  
程序结束

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;

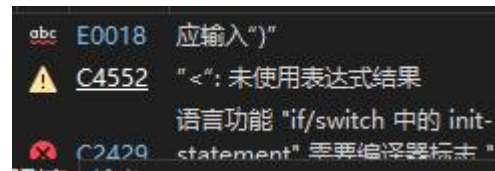
int main()
{
    int i;

    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





## §. 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;
}
```

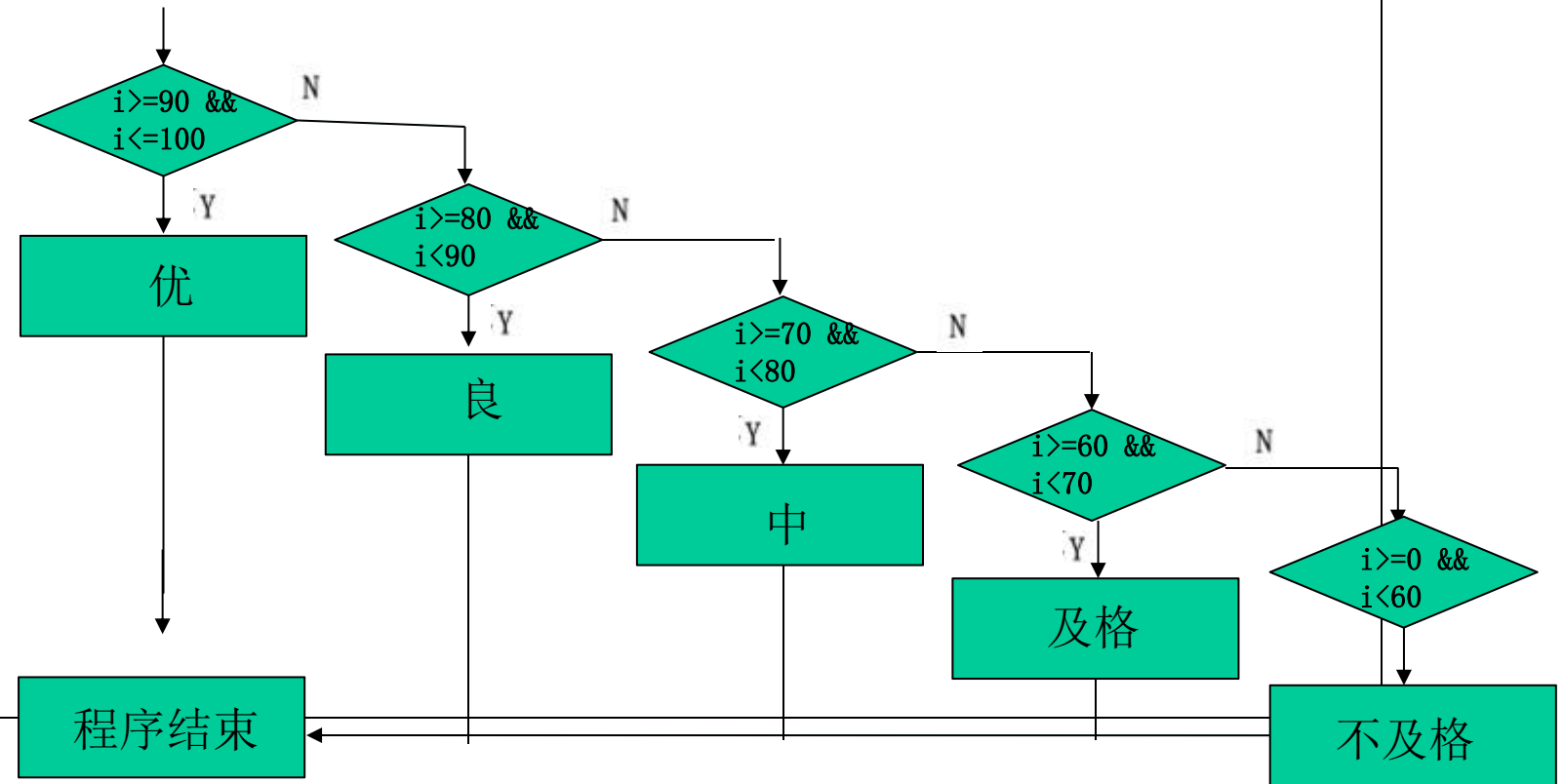
1、给出程序的流程框图(注意字体的清晰可辨)

2、 $i < 90$  能否改为  $i \leq 89$ ? 哪个更好?

可以, 但  $i < 90$  更好

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

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) {	第0行的"{" 和 第5行的"}"配对
1: if (表达式) {	第1行的"{" 和 第2行的"}"配对
2: }	第3行的"{" 和 第4行的"}"配对
3: else {	第6行的"{" 和 第9行的"}"配对
4: }	第7行的"{" 和 第8行的"}"配对
5: }	
6: else {	
7: if (表达式) {	
8: }	
9: }	
	总结：给出大括号配对的基本准则 当识别左括号时跳过，遇见一个右括号立即与上方最近的左括号匹配，再遇见右括号同理。

## §. 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任意(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

<pre>if (表达式1) {     if (表达式2) {         A;     }     else {         B;     }     C; } else {     if (表达式3) {         D;     }     E; }</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> <p>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</p> <p>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</p>
---	---



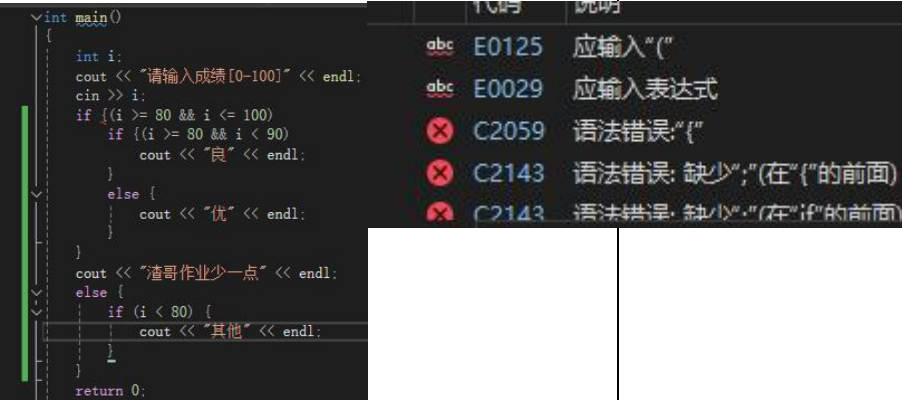
# §. 基础知识题 - 关系运算、逻辑运算与选择结构

6、if语句 - 多重嵌套  
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  
if结束后执行F语句，else无匹配对象

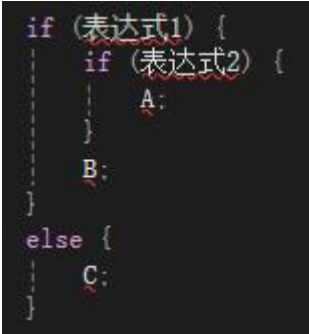
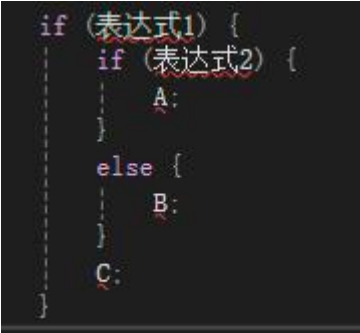




§. 基础知识题 – 关系运算、逻辑运算与选择结构

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> 	<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> 	<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;

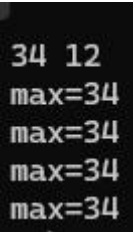
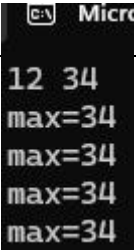
    a > b ? cout << "max=" << a << endl : cout << "max=" << b << endl; //1

    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 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)

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

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

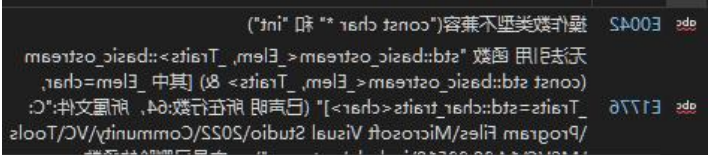
    a==1 ? "Hello" : 123; //编译报错

    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)  
表达式1、2、3的类型可以不同，但2、3的类型必须相同

## §. 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

## 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)  
-9~-1和101~109





# §. 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 &lt;iostream&gt; using namespace std; int main() {     const int k=5;     int score;     cout&lt;&lt;"请输入成绩[0-100]"&lt;&lt;endl;     cin &gt;&gt; score;     switch(score/10) {         case 10:         case 9:             cout&lt;&lt;"优"&lt;&lt;endl;             break;         case 6:             cout&lt;&lt;"及格"&lt;&lt;endl;             break;         default:             cout&lt;&lt;"输入错误"&lt;&lt;endl;             break;         case k+2:             cout&lt;&lt;"中"&lt;&lt;endl;             break;         case 8:             cout&lt;&lt;"良"&lt;&lt;endl;             break;         case 5:         case 4:         case 3:         case 2:         case 1:         case 0:             cout&lt;&lt;"不及格"&lt;&lt;endl;             break;     }      return 0; }</pre>	<p>在8. A的基础上</p> <ul style="list-style-type: none"><li>1、将6、8、default的位置进行了交换</li><li>2、将7写为常变量+常量形式</li></ul> <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>完全一致</p> <p>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</p>
--	--



# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 8、switch-case语句

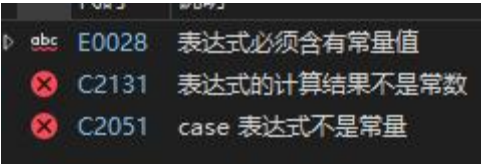
C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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

int k为变量，而case需要表达式为常量



# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 8、switch-case语句

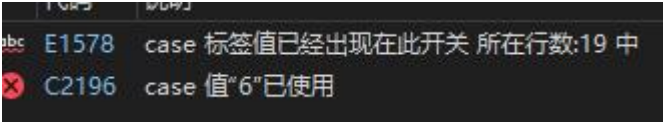
D. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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  
不能出现两个相同的case值





# §. 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)

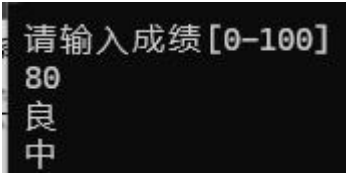
<pre>#include &lt;iostream&gt; using namespace std; int main() {     float score;     cout&lt;&lt;"请输入成绩[0-100]"&lt;&lt;endl;     cin &gt;&gt; score;     switch(score/10) {         case 10:         case 9:             cout&lt;&lt;"优"&lt;&lt;endl;             break;         case 8:             cout&lt;&lt;"良"&lt;&lt;endl;             break;         case 7:             cout&lt;&lt;"中"&lt;&lt;endl;             break;         case 6:             cout&lt;&lt;"及格"&lt;&lt;endl;             break;         case 5:         case 4:         case 3:         case 2:         case 1:         case 0:             cout&lt;&lt;"不及格"&lt;&lt;endl;             break;         default:             cout&lt;&lt;"输入错误"&lt;&lt;endl;             break;     }      return 0; }</pre>	<p>在8.A的基础上，将score从int改为float</p> <p>1、给出编译错误的截图</p> <p>2、解释错误原因</p> <p>case所需的类型必须是整型而不能是float型</p> <div><p>表达式必须包含整数或枚举类型</p><p>此常量表达式的类型为 "int", 而所需类型为 "float"</p><p>此常量表达式的类型为 "int", 而所需类型为 "float"</p><p>此常量表达式的类型为 "int", 而所需类型为 "float"</p><p>此常量表达式的类型为 "int", 而所需类型为 "float"</p></div>
---	---



# §. 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 &lt;iostream&gt; using namespace std; int main() {     int score;     cout&lt;&lt;"请输入成绩[0-100]"&lt;&lt;endl;     cin &gt;&gt; score;     switch(score/10) {         case 10:         case 9:             cout&lt;&lt;"优"&lt;&lt;endl;             break;         case 8:             cout&lt;&lt;"良"&lt;&lt;endl;          case 7:             cout&lt;&lt;"中"&lt;&lt;endl;             break;         case 6:             cout&lt;&lt;"及格"&lt;&lt;endl;             break;         case 5:         case 4:         case 3:         case 2:         case 1:         case 0:             cout&lt;&lt;"不及格"&lt;&lt;endl;             break;         default:             cout&lt;&lt;"输入错误"&lt;&lt;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 停止，防止其连续执行下一语句</p> 
---	--

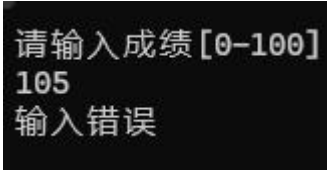




# §. 基础知识题 - 关系运算、逻辑运算与选择结构

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

```
#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;
    cout << "程序结束" << endl;
    return 0;
}
```

- 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）  
写101个case语句
- 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  
if可以，switch不行，因为case必须为整形常量
- 4、总结switch语句使用时的注意事项  
数据最好为10的倍数，case必须赋值整形常量，case之间注意用break分隔
- 5、switch-case语句能完全取代if-else吗？  
综合上面的问题看显然不能，否则程序会变得极其复杂。

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