

# §. 基础知识题 - 关系运算、逻辑运算与选择结构



要求:

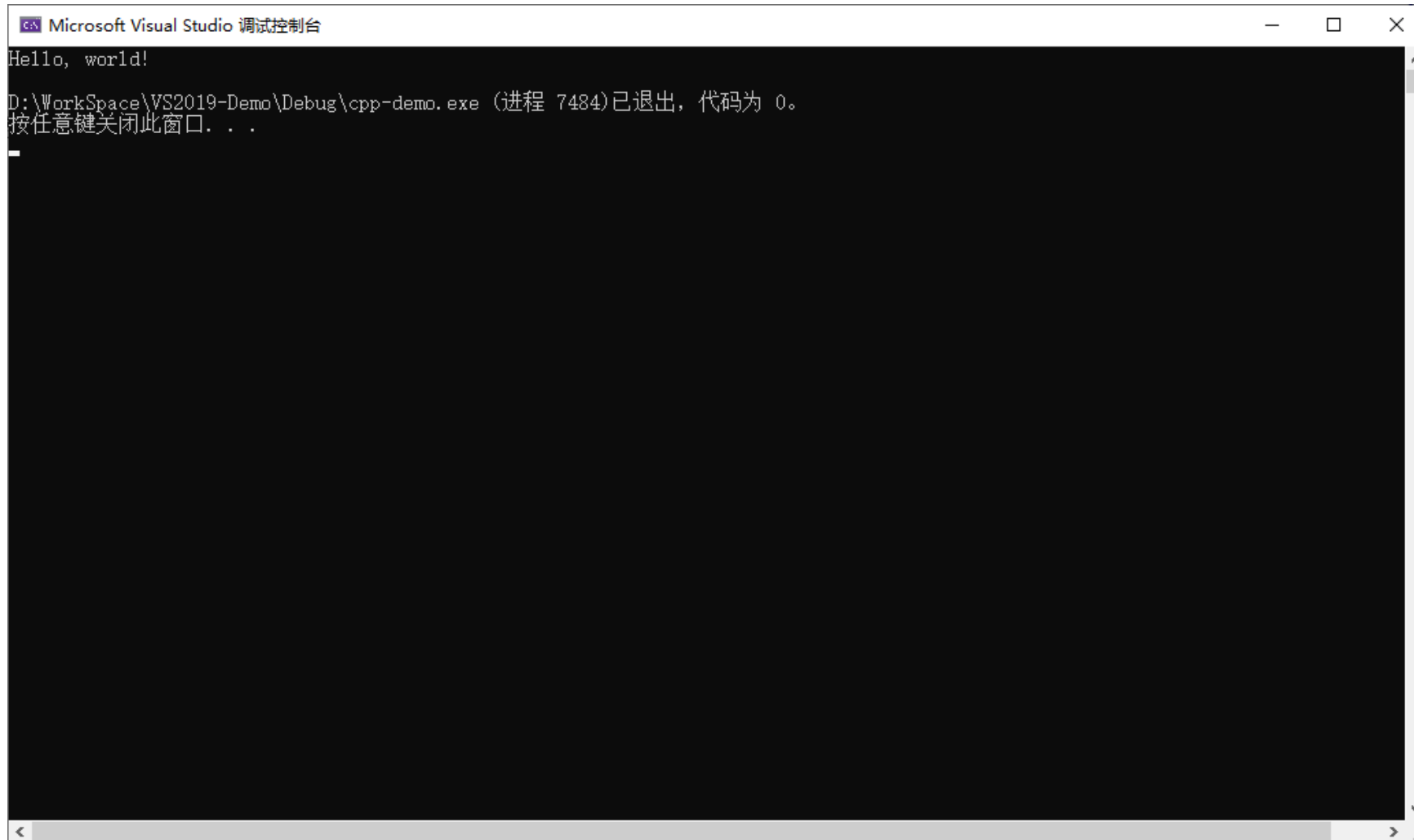
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
  - ★ 贴图要有效部分即可，不需要全部内容
  - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
  - ★ **不允许**手写在纸上，再拍照贴图
  - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
  - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**10月12日前**网上提交本次作业（在“文档作业”中提交）



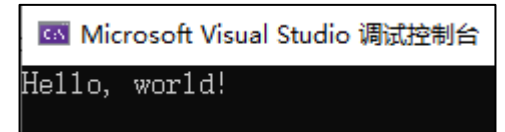
## §. 基础知识题 - 关系运算、逻辑运算与选择结构

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



例：有效贴图

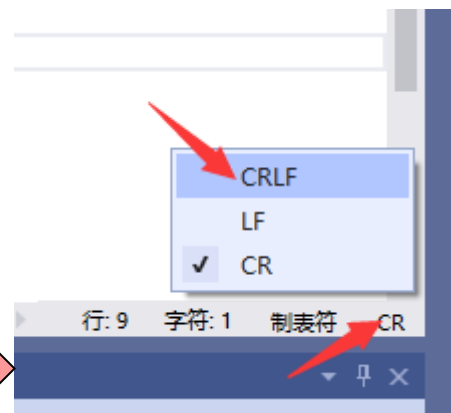




## §. 基础知识题 - 关系运算、逻辑运算与选择结构

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，课件上都有，好好看课件!!!



## §. 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

0  
1  
1

2、VS下为什么会有三个warning? 说说你的理解  
将bool赋值给int类型不安全。



## §. 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

0  
1  
1

2、 $a > b > c$ 这个式子，按常规理解， $3 > 2 > 1$ 是正确的，为什么结果是0？  
 $3 > 2$ 先计算为1然后1在和1进行比较得出0.

$a < b < c$ 这个式子，按常规理解， $3 < 2 < 1$ 是错误的，为什么结果是1？  
先计算 $3 < 2$ ，为0然后 $0 < 1$ 得出1

$b > a < c$ 这个式子，按常规理解， $2 > 3 < 1$ 是错误的，为什么结果是1？  
 $b > a$ ，算出来为0，然后 $0 < c$ ，计算出来为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 \_\_\_\_在数据类型的有效位数范围内比较。 \_\_\_\_

VS

```
0
1
1
4.76837e-08
0
1
```

```
0
1
1
4.76837e-08
0
1
```

dev

```
0
1
1
4.76837e-08
0
1
```

[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

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

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？

前者相同，后者不同。

Float字节数少，只能表示6到7位的有效数字

而double可以表示15到16位的有效数字，

Double更精准可以区分出f1和f2的区别



## §. 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

```
1
true
0
false
```

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

前者是bool类型值为1，后者是字符串

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)

使用boolalpha输出为bool类型


```
cout << "the output is bool(use boolalpha) "
<< boolalpha << test << endl;
```



# §. 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)

<pre>#include &lt;iostream&gt; using namespace std;  int main() {     bool k1 = true;      cout &lt;&lt; sizeof(true) &lt;&lt; endl;     cout &lt;&lt; sizeof(k1) &lt;&lt; endl;     cout &lt;&lt; k1 &lt;&lt; ' ' &lt;&lt; int(k1) &lt;&lt; endl;      cout &lt;&lt; endl;      bool k2 = false;     cout &lt;&lt; sizeof(false) &lt;&lt; endl;     cout &lt;&lt; sizeof(k1) &lt;&lt; endl;     cout &lt;&lt; k2 &lt;&lt; ' ' &lt;&lt; int(k2) &lt;&lt; endl;      return 0; }</pre>	<div>1、贴运行结果</div>  <div>2、bool型常量/变量在内存中占用__1_字节，值是__0/1__</div> <div>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) 输出时按整型量进行处理</div>
---	--



## §. 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;

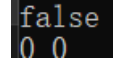
    return 0;
}
```

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：

输入时按照非零为真零为假的原则



## §. 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 0
```

2、解释VS下warning的意思

位数高的数据类型赋值给bool字节数低的数据类型时会存在截断

3、k='A' b1字节赋值给1字节，为什么还有warning?

Bool的赋值遵循非零为真零为假的原则，所以在赋值时也有可能不能保证把k的值正确赋给bool类型

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)

0 现在是1 Bool的赋值遵循非零为真零为假的原则

5、为什么不 cout << c, 而是 (int)c ?

直接cout<<c, 会输出字符，而不是数字

6、“非0为真0为假”这句话如何解释?

赋值给bool类型的值，如果原来的值非零，那么bool类型的值为1，否则为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

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

11

2、当bool参与表达式计算时，当做\_\_\_\_int类型的0或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、 $a>b = 0$ ;

2、 $m=0$ ;

3、结束，整个表达式为0

3、短路运算的意思是：\_仅当必须执行下一个逻辑运算符才能求出解时，才执行该运算符，否则不执行

---



## §. 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=1, n=1;

    cout << "m=" << m << " n=" << n << endl;
    m=5>3 && 2 || n=8<4 - !0;
    cout << "m=" << m << " n=" << n << endl;

    return 0;
}
```

有以下逻辑表达式

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

1、构造一个测试程序，在不改变该表达式目前求值顺序的情况下（允许插入新的运算，但目前这几个运算符的顺序不要变），证明  $8<4 - !0$  存在短路运算

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 $< - !$ 运算符。（所有 / 比 $||$  优先级高的）

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

int main()
{
    int m = 0, n = 1;

    cout << "m=" << m << " n=" << n << endl;
    (m=5>3) && 2 || (n=8<4) - !0;
    cout << "m=" << m << " n=" << n << endl;

    Microsoft Visual Studio 调试控制台
    m=0 n=1
    m=1 n=1
```



## §. 基础知识题 – 关系运算、逻辑运算与选择结构

此页不要删除，也没有意义，仅仅为了分隔题目



## §. 基础知识题 - 关系运算、逻辑运算与选择结构

### 5、if语句 - 基本使用

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

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

int main()
{
    int i;

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

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

    return 0;
}
```

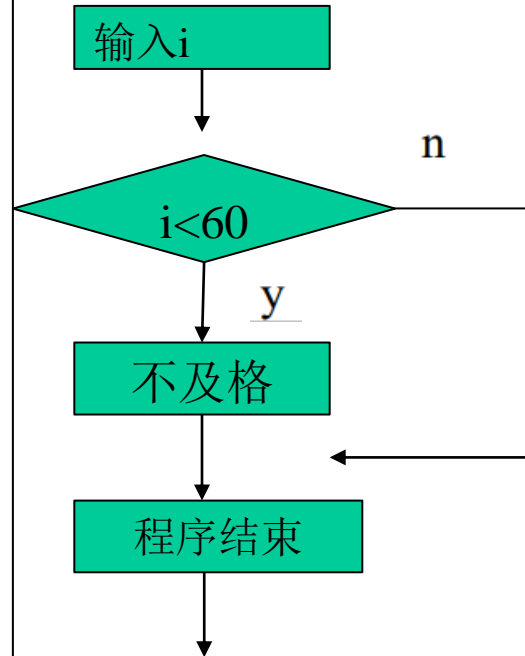
1、输入34，贴运行结果

```
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
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

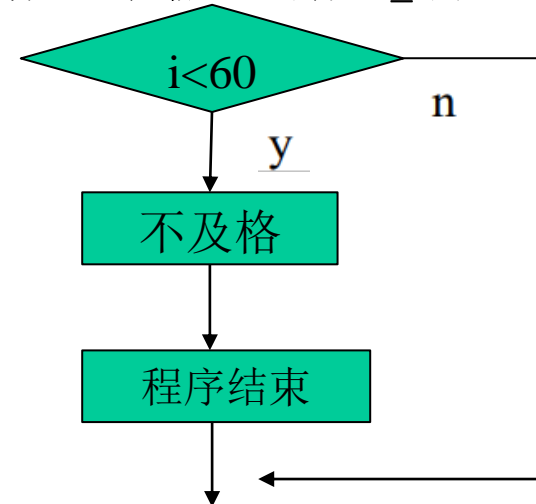
```
请输入成绩[0-100]
34
不及格
程序结束
```

2、输入74，贴运行结果

```
请输入成绩[0-100]
74
```

3、画出程序流程图

4、程序标注“未缩进”的行，\_不应该\_\_\_\_\_（应该/不应该）缩进





# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 5、if语句 - 基本使用

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

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

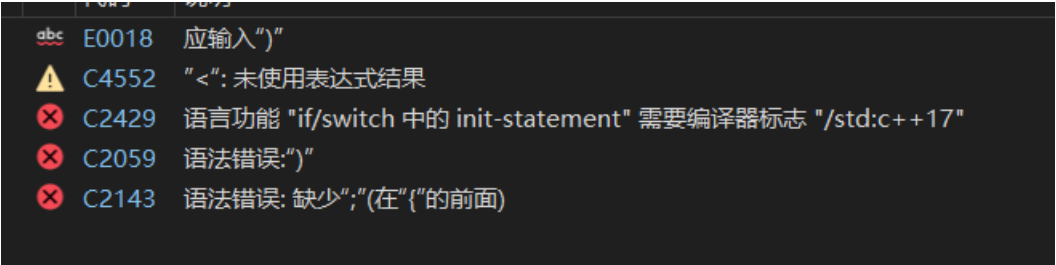
int main()
{
    int i;

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

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

    return 0;
}
```

贴编译错误并给出解释  
If括号里面加了分号





# §. 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)

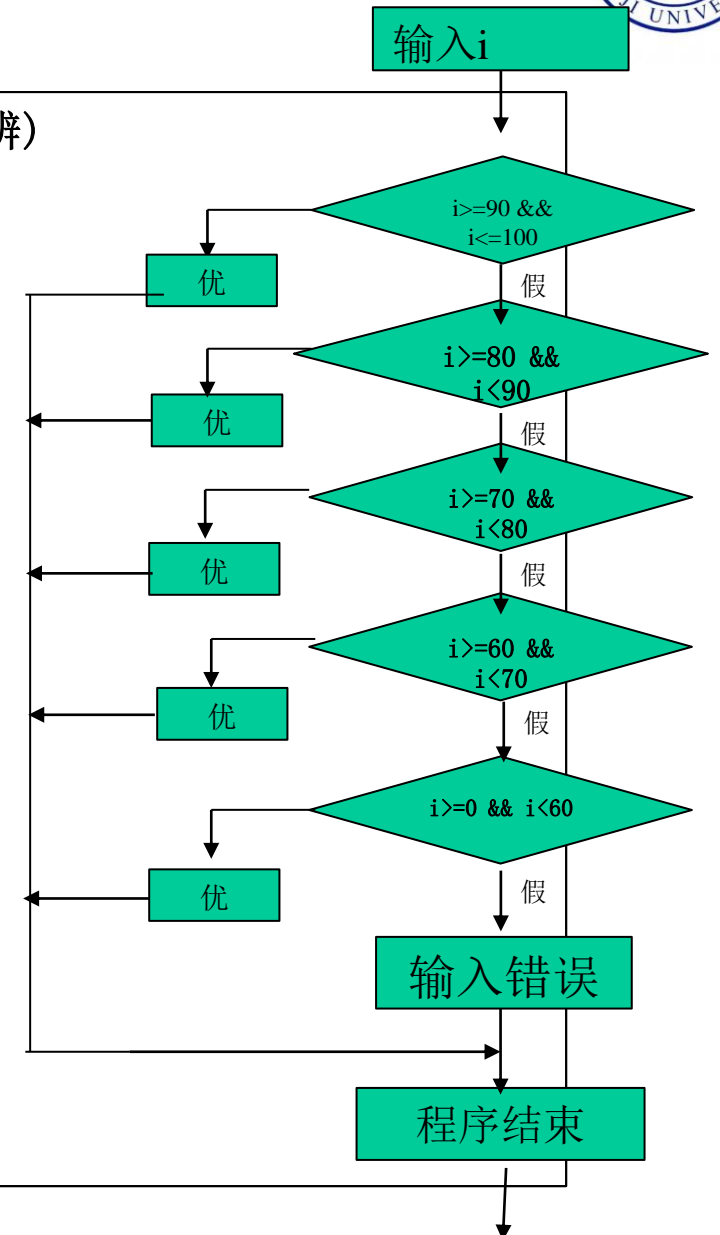
2、i<90能否改为i<=89? 哪个更好?

可以，有小数不行；前者好

3、i<90能否改为i<=90? 运行是否正确?

运行正确

不及格







## §. 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
---	---



## §. 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

```
#include <iostream>  
using namespace std;  
int main()  
{  
    if (1) {  
        if (2) {  
            cout << 'A';  
        }  
        else {  
            cout << 'B';  
        }  
        cout << 'C';  
    }  
    cout << 'F';  
    else {  
        if (表达式3) {  
            cout << "D";  
        }  
        cout << "E";  
    }  
}
```

abc E0127 应输入一个语句

Project1



# §. 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>	左侧代码按缩进格式排版	<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__真__(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>	左侧代码按缩进格式排版	<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

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

2、输入34 12，给出运行截图

```
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 1 3\_\_





## §. 基础知识题 – 关系运算、逻辑运算与选择结构

### 7、条件运算符与条件表达式

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

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

```
int main()
{
```

```
    int a=1, b=2;
```

```
    a==1 ? "Hello" : 123; //编译报错
```

```
    a>b ? cout << a : printf("%d", b); //编译报错
```

```
    a==1 ? 'A' : 123; //编译正确
```

```
    return 0;
```

```
}
```

1、给出编译报错的截图

2、条件表达式使用的三句中，前两句报错，最后一句正确，总结下条件表达式使用时的限制规则  
(提示：注意表达式2和表达式3的类型)  
表达式中第二个和第三个参数类型要相容

```
abc E0042 操作数类型不兼容("const char *" 和 "int")
无法引用 函数 "std::basic_ostream<_Elem, _Traits>::basic_ostream(const std::basic_ostream<_Elem, _Traits> &)" [其中 _Elem=char,
abc E1776 _Traits=std::char_traits<char>]" (已声明 所在行数:65, 所属文件:"F:\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.37.32822\include
\ostream") -- 它是已删除的函数
abc E0042 操作数类型不兼容("std::basic_ostream<char, std::char_traits<char>>" 和 "int")
✗ C2446 "(": 没有从"int"到"const char [6]"的转换
✗ 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)
```



## §. 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

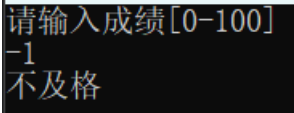
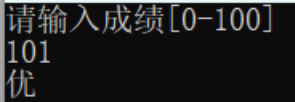
A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#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)

<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> <ol 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></ol> <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_完全一致_____ (完全一致/不完全一致) 如果不完全一致，给出表现不一致的测试数据的运行截图</p>
--	---



# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 8、switch-case语句

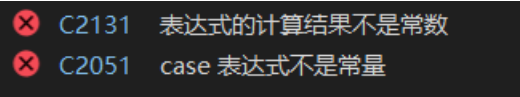
C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

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

    return 0;
}
```

在8.B的基础上，将k从const int改为int

1、给出编译错误的截图



2、解释错误原因

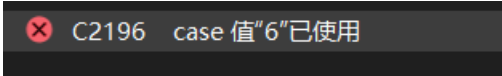
Case中的表达式一定要是整型常量或者整型的常量表达式



# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 8、switch-case语句

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

<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:         case 4+2:             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，多了一个case 4+2</p> <p>1、给出编译错误的截图</p>  <p>2、解释错误原因</p> <p>Case中不能出现相同的值</p>
---	--



# §. 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

abc E2373 此常量表达式的类型为 "int", 而所需类型为 "float"

abc E2373 此常量表达式的类型为 "int", 而所需类型为 "float"

abc E2373 此常量表达式的类型为 "int", 而所需类型为 "float"

abc E2373 此常量表达式的类型为 "int", 而所需类型为 "float"

abc E2373 此常量表达式的类型为 "int", 而所需类型为 "float"

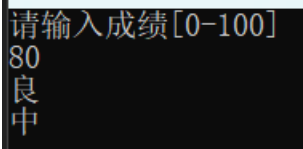
Case中的表达式一定要是整型常量或者整型的常量表达式



# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 8、switch-case语句

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 退出switch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)

```
#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

```
#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)
        score = 110;
    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;
}
```

```
请输入成绩[0-100]
101
输入错误
```



## §. 基础知识题 – 关系运算、逻辑运算与选择结构

### 8、switch-case语句

#### H. 思考

如果将成绩区间对应为：

[85-100]	- 优
[70-85)	- 良
[60-70)	- 及格
[0-60)	- 不及格

1、用if-else语句完成该程序并贴图

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）

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  
都能用，整除运算符会忽略小数

4、总结switch语句使用时的注意事项

每个case最后要加break，注意输入只能为整数

5、switch-case语句能完全取代if-else吗？  
不能

```
#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)
        score = 110;
    else if (score >= 85 && score <= 100)
        cout << "优" << endl;
    else if (score >= 70 && score <= 85)
        cout << "良" << endl;
    else if (score >= 60 && score <= 70)
        cout << "及格" << endl;
    else
        cout << "不及格" << endl;

    return 0;
}
```

```
int main()
{
    int score;
    cout << "请输入成绩[0-100]" << endl;
    cin >> score;
    if (score > 100 || score < 0)
        score = 110;
    switch ((score+5) / 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:
            cout << "不及格" << endl;
            break;
    }
```



## §. 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