

# §. 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26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 Microsoft Visual Studio 2022 IDE interface. In the top-left, there's a code editor window titled "demo.cpp" containing the following C++ code:

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

In the bottom-right corner of the code editor, there is a status bar showing "行: 4 字符: 2 空格 SYS CR". A red arrow points from this status bar to a small floating menu labeled "CRLF" which contains options "CRLF", "LF", and "CR". The "CR" option is checked.

Below the code editor is the "Output" window. It displays the build log:

```
生成开始于 22:23...
1>—— 已启动生成: 项目: demo-cpp, 配置: Debug Win32 ——
1>demo.cpp
1>D:\WorkSpace\VS2022-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\VS2022-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>MSVCRTD.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) 中引用了该符号
1>D:\WorkSpace\VS2022-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。
===== 生成: 0 成功, 1 失败, 0 最新, 0 已跳过 =====
===== 生成于 22:23 完成, 耗时 01.132 秒 =====
```

A red box highlights the warning message: "1>D:\WorkSpace\VS2022-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".

At the bottom left of the output window, there are two tabs: "错误列表" (Errors List) and "输出" (Output). A red arrow points from the bottom-left corner of the output window towards the "错误列表" tab.

## §. 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



warning C4804: ">" : 在操作中使用类型 "bool" 不安全  
warning C4804: "<" : 在操作中使用类型 "bool" 不安全  
warning C4804: "<" : 在操作中使用类型 "bool" 不安全

### 2、VS下为什么会有三个warning？说说你的理解

在这些表达式中，布尔值被隐式转换为整数，这可能会导致逻辑错误或难以发现的 bug。



## §. 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



```
warning C4804: ">" : 在操作中使用类型 "bool" 不安全
: warning C4804: "<" : 在操作中使用类型 "bool" 不安全
: warning C4804: "<" : 在操作中使用类型 "bool" 不安全
```

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、 $a>b>c$ :首先计算  $a > b$ ，即  $3 > 2$ ，结果为 true (即 1)。然后计算  $1 > c$ ，即  $1 > 1$ ，结果为 false (即 0)。因此，d 被赋值为 0。  
2、 $a<b<c$ :首先计算  $a < b$ ，即  $3 < 2$ ，结果为 false (即 0)。然后计算  $0 < c$ ，即  $0 < 1$ ，结果为 true (即 1)。因此，d 被赋值为 1。  
3、 $b > a < c$ :首先计算  $b > a$ ，即  $2 > 3$ ，结果为 false (即 0)。然后计算  $0 < c$ ，即  $0 < 1$ ，结果为 true (即 1)。因此，d 被赋值为 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

warning C4305: “初始化”：从“double”到“float”截断  
0  
1  
1  
4.76837e-08  
0  
1

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

VS      warning C4305: “初始化”：从“double”到“float”截断

0  
1  
1  
4.76837e-08  
0  
1

Dev      [Error] 'fabs' was not declared in this scope; did you mean 'labs'?

3、由本例得出的结论，实数进行相等比较时的通用方法是由于浮点数的精度问题，直接使用 == 进行相等比较可能会导致不准确的结果。通用方法是使用一个很小的阈值（如 1e-6）来判断两个浮点数是否足够接近。



## §. 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

```
warning C4305: “初始化”：从“double”到“float”截断
warning C4305: “初始化”：从“double”到“float”截断
```

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？  
1e-6 相同，1e-7 不同。

原因：float 类型的精度约为 7 位有效数字，而 double 类型的精度约为 15 位有效数字。因此，float 类型在比较时更容易因为精度不足而认为两个数相等，而 double 类型能够更精确地表示和比较数值。

## §. 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

### 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")

true和false是布尔常量，分别表示逻辑真和逻辑假。在输出时，true会被转换为整数1，false会被转换为0。

"true" 和 "false" 是字符串字面量，输出时会直接显示为字符串 "true" 和 "false"

3、进阶思考：目前直接输出逻辑常量true和false，在屏幕上输出的输出是1/0，如果想输出为true/false，应该怎么做？

**注意：** 1、不允许用分支语句/条件运算符

2、提示：去网上查一个前导格式控制符(课件无)

可以使用boolalpha 格式控制符来改变布尔值的输出格式。



## §. 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

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

2、bool型常量/变量在内存中占用1字节，值是true（表示为1），false（表示为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型常量/变量在无签到控制符时，默认输出为0或1，true输出为1。false输出为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

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：



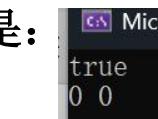
2、输入1，输出是：



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 0 会被输出false，转为0。

输入任何非零值如会输出true，转为1。

输入字符串输出false，转为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;

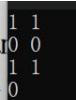
    k='A';
    cout << k << ' ' << (int)k << endl;

    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	warning C4305: “=” : 从 “char” 到 “bool” 截断 warning C4305: “=” : 从 “int” 到 “bool” 截断 : warning C4305: “初始化” : 从 “int” 到 “char” 截断 : warning C4309: “初始化” : 截断常量值	
2、解释VS下waring的意思	某一变量的类型字节数比实际赋值占的字节数少，发生截断。	
3、k='A'是1字节赋值给1字节，为什么还有warning？	虽然 ‘A’ 是一字节，但char型给bool型赋值，由于bool型只能存储0或1，因此还是会提示warning	
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？ <b>(和c对比)</b>	k为false，即k=0，实际为0，当将其赋值给 bool 类型时，只有最低的 8 位（即 00000000）被保留
5、为什么不 cout << c，而是 (int)c？	cout << c 会输出字符 c 的 ASCII 字符表示。为了输出 c 的整数值，需要将其强制转换为 int 类型，即 (int)c。	
6、“非0为真0为假”这句话如何解释？	bool 类型的变量在条件判断中，任何非零值（正数或负数）都被视为 true，而 0 被视为 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0 (false时)，1 (true时)

## §. 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）

步骤 1：计算  $a > b$ ，即  $1 > 2$ ，结果为 false（即 0）。  
步骤 2：将  $a > b$  的结果赋值给  $m$ ，因此  $m$  被赋值为 0。  
步骤 3：由于  $(m=a>b)$  的结果为 false，且  $\&\&$  运算符具有短路特性，因此不会计算  $(n=c>d)$  部分。  
步骤 4： $n$  保持原来的值 1。

3、短路运算的意思是：对于  $\&\&$  运算符，如果左边的表达式为 false，则整个表达式必定为 false，因此不会计算右边的表达式。

对于  $\|$  运算符，如果左边的表达式为 true，则整个表达式必定为 true，因此不会计算右边的表达式。



## §. 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;
}
```

```
{
    int a = 0, b = 1, c = 0;

    cout << "a=" << a << " b=" << b << " c=" << c << endl;
    (a = 5 > 3) && 2 || (b=8 < 4 - !0), c=2<3 ;
    cout << "a=" << a << " b=" << b << " c=" << c << endl;

    return 0;
}
```

Microsoft Visual Studio 调试控制台

```
a=0 b=1 c=0
a=1 b=1 c=1
```

D:\XUE\CTF\VS2022\homework\Debug\模块3 part5.exe (进和

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 $\text{||}$ ），忽略比 $\text{||}$ 优先级高的运算符。（所有 / 比 $\text{||}$ 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;

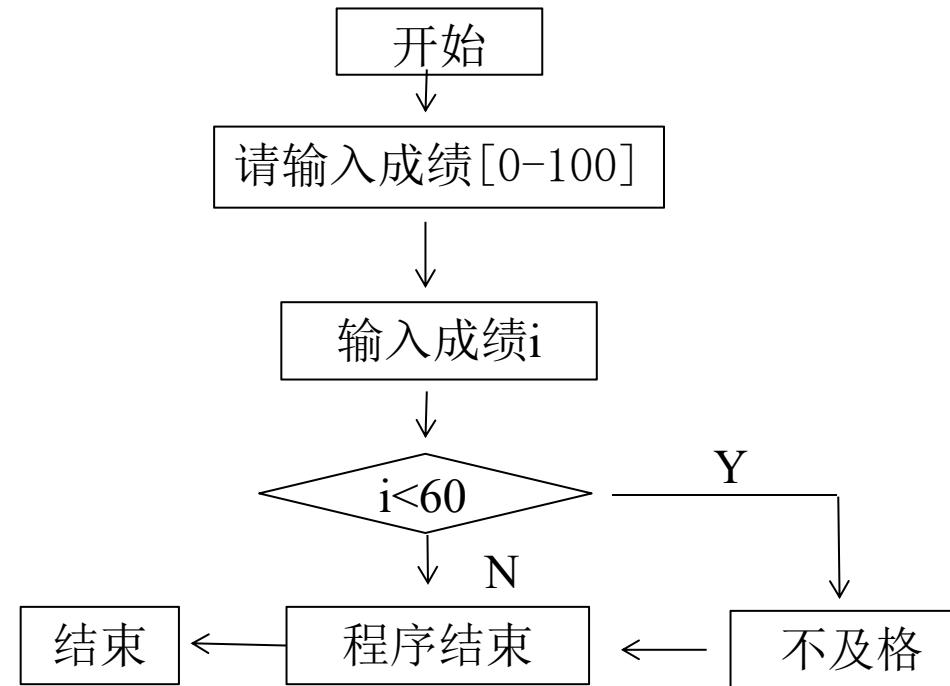
    if (i<60) {
        cout << "不及格" << endl;
    }
    cout << "程序结束" << endl;

    return 0;
}
```

1、输入34，贴运行结果

2、输入74，贴运行结果

3、画出程序对应的流程框图





## §. 基础知识题 - 关系运算、逻辑运算与选择结构

### 5. if语句 - 基本使用

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

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

int main()
{
    int i;

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

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

    return 0;
}
```

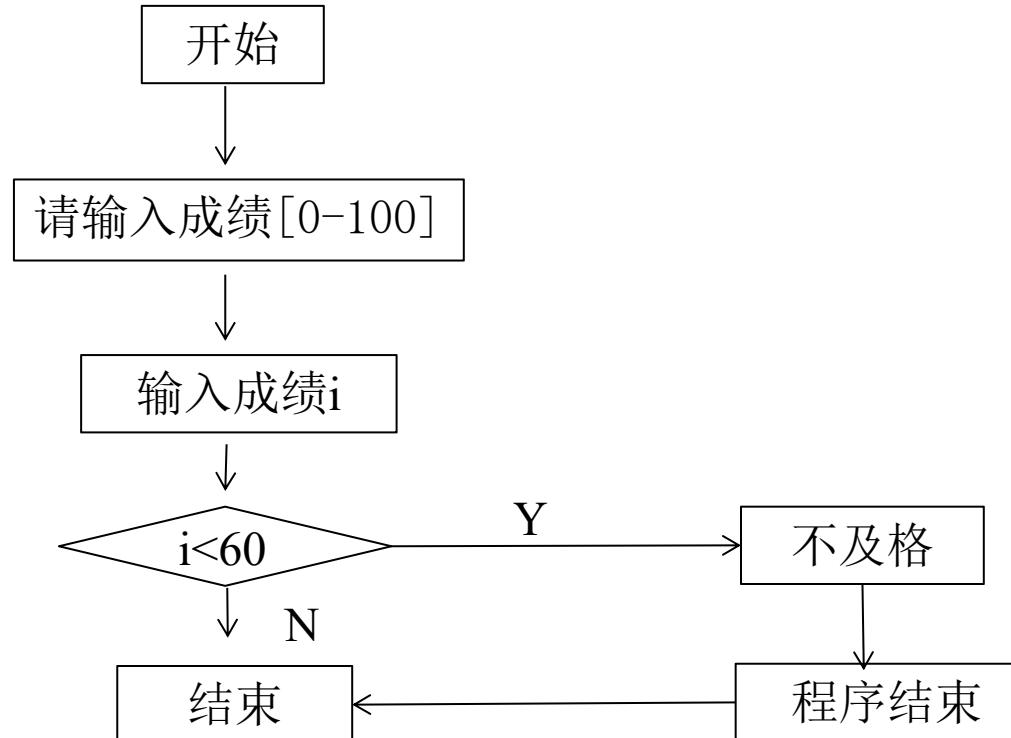
1、输入34，贴运行结果

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

```
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)
```

解释：;会直接结束一句话，截断了“(i<60”和“)”导致语法出现错误。



## §. 基础知识题 - 关系运算、逻辑运算与选择结构

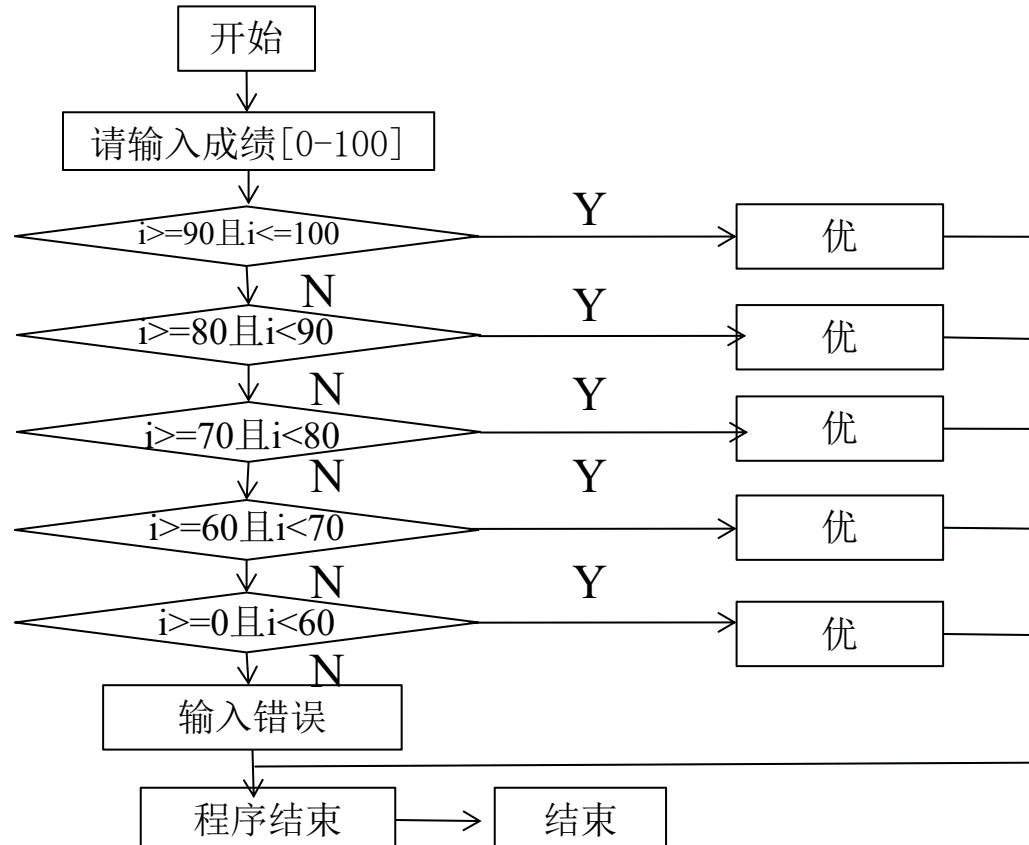
### 5. if语句 - 基本使用

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?

能。 $i < 90$ 更好，因为它更直观地表达了“小于 90”的条件。

3.  $i < 90$ 能否改为 $i \leq 90$ ? 运行是否正确? 不能, 不正确。

## §. 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

就近匹配原则：每个 { 会与最近的 } 配对。

嵌套规则：内层的 { 和 } 必须完全包含在外层的 { 和 } 中。

缩进对齐：通过缩进可以清晰地看出 { 和 } 的配对关系，避免混淆。

避免遗漏：每个 { 必须有一个对应的 }，否则会导致编译错误。



## §. 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

error C2181: 没有匹配 if 的非法 else

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int i;  
    cin >> i;  
    if (i>=90) {  
        if (i==100) {  
            cout << "恭喜你考了满分！" << endl;  
        }  
        else {  
            cout << "你得到了优秀！";  
        }  
        cout << "非常好！" << endl;  
    }  
    cout << "加油" << endl;  
    else {  
        if (i<60) {  
            cout << "你没有及格。" << endl;  
        }  
        cout << "还需要努力。" << endl;  
    }  
    return 0;  
}
```

2、请说明错误原因（如果计算机的报错和你的认知不同，想明白为什么）  
错误原因是 else 没有正确匹配到 if，被F语句截断了。



## §. 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>	<pre>左侧代码按缩进格式排版 if (表达式1) {     if (表达式2) {         A;     }     B; } else {     C; }</pre>	<ol style="list-style-type: none"><li>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</li><li>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</li><li>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</li></ol>
<pre>if (表达式1) { if (表达式2) { A; } else { B; } C; }</pre>	<pre>左侧代码按缩进格式排版 if (表达式1) {     if (表达式2) {         A;     }     else {         B;     }     C; }</pre>	<ol style="list-style-type: none"><li>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</li><li>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</li><li>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</li></ol>

## §. 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;

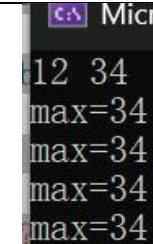
    if (a>b)
        cout << "max=" << a << endl;
    else
        cout << "max=" << b << endl;

    a > b ? cout << "max=" << a << endl : cout << "max=" << b << endl; //1
    cout << "max=" << (a>b?a:b) << endl; //2

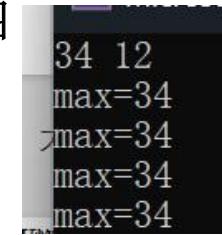
    printf("max=%d", a>b?a:b); //3

    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3, 2, 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

error C2446: “:” : 没有从“int”到“const char [6]”的转换

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)

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)

条件表达式中，表达式1 和 表达式2 的类型必须兼容，否则会导致编译错误。同时表达式1 和 表达式2 不能使用两种语言。

## §. 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

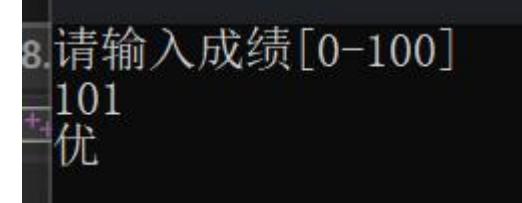
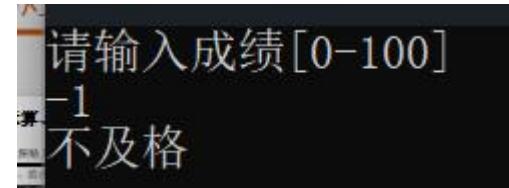
### 8. switch-case语句

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)

```
#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

error C2131: 表达式的计算结果不是常数

error C2051: case 表达式不是常量

2、解释错误原因

switch 语句的 case 标签必须是常量表达式。case后面的值必须在编译时确定，不能是变量或运行时计算的表达式。

由于 k 是变量， $k + 2$  是一个运行时计算的表达式，而不是编译时常量。因此报错。



# §. 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

error C2196: case 值“6”已使用

2、解释错误原因

同一个switch中，case后面的值不能相同。



# §. 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)

```
#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 error C2450: 类型为“float”的 switch 表达式无效

2、解释错误原因

switch 语句的表达式必须是整数类型，不能是浮点数类型。



# §. 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

请输入成绩[0-100]  
80  
良  
中

2、解释break的作用

终止当前的 case 块，避免贯穿到下一个 case 块。



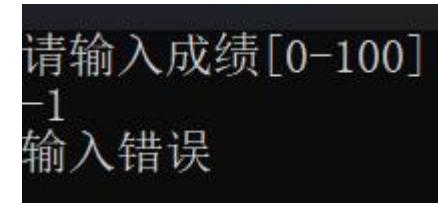
# §. 基础知识题 - 关系运算、逻辑运算与选择结构

## 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;
    if (score < 0 || score > 100) {
        cout << "输入错误" << endl;
        return 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;
        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



# §. 基础知识题 - 关系运算、逻辑运算与选择语句

## 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）

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可以处理浮点类型，能；switch只能处理整型，不能。

4、总结switch语句使用时的注意事项

switch 的表达式必须是整数类型，case 标签必须是常量表达式，  
每个 case 块通常以 break 结束，否则会“贯穿”到下一个 case。

5、switch-case语句能完全取代if-else吗？

switch适用于离散值的比较，而 if-else 适用于任意条件表达式  
switch无法处理复杂的逻辑条件（如score>= 84&&score<=100）。

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

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

int main() {
    float score;
    cout << "请输入成绩[0-100]: " << endl;
    cin >> score;

    if (score >= 84 && score <= 100) {
        cout << "优" << endl;
    }
    else if (score >= 68 && score < 84) {
        cout << "良" << endl;
    }
    else if (score >= 55 && score < 68) {
        cout << "及格" << endl;
    }
    else if (score >= 0 && score < 55) {
        cout << "不及格" << endl;
    }
    else {
        cout << "输入错误" << endl;
    }

    return 0;
}
```

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

int main() {
    int score;
    cout << "请输入成绩[0-100]: " << endl;
    cin >> score;
    if (score < 0 || score > 100) {
        cout << "输入错误" << endl;
        return 0;
    }

    switch (score / 10) {
        case 10:
        case 9:
        case 8:
            if (score >= 84) {
                cout << "优" << endl;
            }
            else {
                cout << "良" << endl;
            }
            break;
        case 6:
        case 5:
            if (score >= 55) {
                cout << "及格" << endl;
            }
            else {
                cout << "不及格" << endl;
            }
            break;
        default:
            cout << "不及格" << endl;
            break;
    }

    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