



1、正常顺序结构程序

- 要求：(1) 函数内所有语句顺序缩进四格
(2) 一条语句一行，左右括号单独一行
(3) 适当加入注释、空行

<pre>#include <iostream> using namespace std; int main() { int a; a=10; cout << a << endl; return 0; }</pre>	<pre>#include <iostream> using namespace std; int main() { int a; ← 不要这样 a=10; cout << a << endl; return 0; }</pre>	<pre>#include <iostream> using namespace std; int main() { ← 不要这样 int a; a=10; cout << a << endl; return 0; }</pre>
--	---	---

- (4) 若某行过长，可以分开数行，左侧对齐/缩进均可

<pre>//第2行对齐 cout << "样例第一行" << "样例第二行";</pre>	<pre>//第2行缩进 cout << "样例第一行" << "样例第二行";</pre>
--	--

★ 不符合要求者，每题扣10%的格式分



2、含有单分支结构的程序

要求：(1) 分支内的语句顺序缩进四格

(2) 左大括号两种方式(if后/另起一行)任选；右大括号单独一行(可缩进/不缩进)

(3) 分支内只有一个语句的，可以不要大括号，但必须另起一行

<pre>if (....) 语句1;</pre>	<pre>if (....) { ←不缩进 语句1; ... 语句n; } ←不缩进</pre>	<pre>if (....) { 语句1; ... 语句n; } ←不缩进</pre>	<pre>if (....) { 语句1; ... 语句n; } ←缩进</pre>
<pre>if (....) 语句1;</pre>		<pre>if (....) { 多个语句 }</pre>	
不允许		不允许	



3、含有双分支结构的程序

要求：(1) 分支内的语句顺序缩进四格

(2) 左大括号两种方式(if后/另起一行)任选；右大括号单独一行(可缩进/不缩进)

(3) 分支内只有一个语句的，可以不要大括号，但必须另起一行

<pre>if (....) 语句1; else 语句2;</pre>	<pre>if (....) { ←不缩进 语句1; ... 语句n; } ←不缩进 else { ←不缩进 语句1; ... 语句n; } ←不缩进</pre>	<pre>if (....) { 语句1; ... 语句n; } ←不缩进 else { 语句1; ... 语句n; } ←不缩进</pre>	<pre>if (....) { 语句1; ... 语句n; } ←缩进 else { 语句1; ... 语句n; } ←缩进</pre>
<pre>if (....) 语句1; else 语句2;</pre>	不允许		<pre>if (....) { 多个语句 } else { 多个语句 }</pre> 不允许



4、含有多分支结构的程序

要求：(1) 分支内的语句顺序缩进四格

(2) 左大括号两种方式(if后/另起一行)任选；右大括号单独一行(可缩进/不缩进)

(3) 分支内只有一个语句的，可以不要大括号，但必须另起一行

<pre>if (...) 语句1; else if (...) 语句2; else 语句x;</pre>	<pre>if (...) { ← 不缩进 语句1; ... 语句n; } ← 不缩进 else if (...) { 语句1; ... 语句n; } else { ← 不缩进 语句1; ... 语句n; } ← 不缩进</pre>	<pre>if (...) { 语句1; ... 语句n; } ← 不缩进 if (...) { 语句1; ... 语句n; } ← 不缩进 else { 语句1; ... 语句n; } ← 不缩进</pre>	<pre>if (...) { 语句1; ... 语句n; } ← 缩进 if (...) { 语句1; ... 语句n; } ← 缩进 else { 语句1; ... 语句n; } ← 缩进</pre>
<pre>if (...) 语句1; else if (...) 语句2; else 语句x;</pre> <p>else if 写在一行即可， 不建议下面这种形式， 但不扣分</p>			
<pre>if (...) 语句1; else if (...) 语句2; else 语句x;</pre>	不允许	<pre>if (...) { 多个语句 } else if (...) { 多个语句 } else { 多个语句 }</pre>	不允许



5、含有分支相互嵌套的程序

要求：(1) 每层均需顺序缩进四格

(2) 左大括号两种方式(if后/另起一行)任选；右大括号单独一行(可缩进/不缩进)

<pre>if (....) { ← 不缩进 if (...) { 语句1; ... 语句n; } ... 语句n; } ← 不缩进</pre>	<pre>if (....) { if (...) { 语句1; ... 语句n; } ← 不缩进 ... 语句n; } ← 不缩进</pre>	<pre>if (....) { if (...) { 语句1; ... 语句n; } ← 缩进 ... 语句n; } ← 缩进</pre>
--	--	--



6、含有switch/case的语句

要求：(1) case比switch缩进4格，单独一行，后面不要再跟语句

(2) case下的语句比case再缩进4格

<pre>switch(x) { case 1: cout << a; break; case 2: ...; default: ...; break; } ← 不缩进</pre>	<pre>switch(x) { case 1: cout << a; break; case 2: ...; default: ...; break; } ← 不缩进</pre>	<pre>switch(x) { case 1: cout << a; break; case 2: ...; default: ...; break; } ← 缩进</pre>
--	--	---

不允许写成 case 1:cout << a; break;

★ 其它未列举的、更复杂的嵌套组合，以4个空格为单位，按层次依次缩进，每层比上层缩进四格

★ 不符合要求者，每题扣10%的格式分

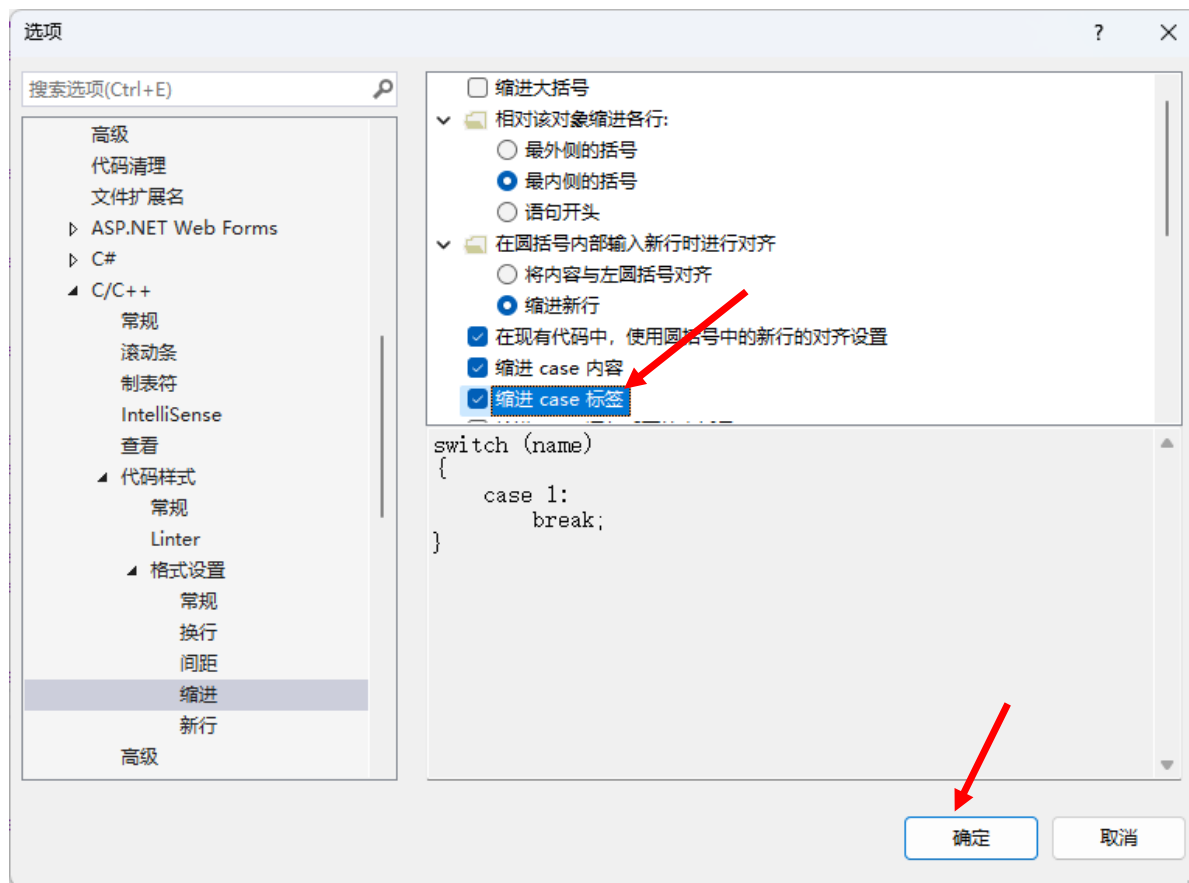


附：VS2022中缺省格式，case不会比switch缩进四格，需要设置，具体方法为：

“工具”菜单 - “选项” - 出现“选项”对话框

- 文本编辑器 - C/C++ - 代码样式 - 格式设置 - 缩进

- 选中“缩进case标签” - 按“确定”即可



注：其它设置项也可以自行研究



7、含有循环结构的程序(while和for)

要求：(1) 循环内的语句顺序缩进四格

(2) 右大括号单独占一行，左大括号两种方式任选

(3) 循环内只有一个语句的，也要另起一行

(4) 循环内是空语句的，分号另起一行

<pre>while (...) { 语句1; ... 语句n; } ←不缩进</pre>	<pre>for (...) { 语句1; ... 语句n; } ←不缩进</pre>	<pre>for (...) { 语句1; ... 语句n; } ←缩进</pre>
<pre>while (...) 语句;</pre>	<pre>while(...) 语句;</pre>	<pre>while (...) ;</pre>
<pre>for (...) 语句;</pre>	<pre>for (...) 语句;</pre>	<pre>for (...) ;</pre>
允许	不允许	当空语句时 允许
		<pre>while (...); for (...);</pre> 不允许



8、含有循环结构的程序(do-while)

要求：(1) 右括号不缩进，后面直接跟while

(2) 左括号两种方式（直接在do后面/下一行）均可

(3) 只有一个语句也加括号

<pre>do { 语句1; ... 语句n; } while(...);</pre>	<pre>do { 语句1; ... 语句n; } while(...);</pre>	<pre>do { 语句1; } while(...);</pre>
---	---	--



9、含有循环相互嵌套的程序

要求：(1) 每层均需顺序缩进四格

(2) 右大括号单独占一行，左大括号两种方式任选

<pre>while (....) { for (...) { 语句1; ... 语句n; } ←不缩进 ... 语句n; } ←不缩进</pre>	<pre>for (....) { while (...) { 语句1; ... 语句n; } ←不缩进 ... 语句n; } ←不缩进</pre>	<pre>for (....) { while (...) { 语句1; ... 语句n; } ←缩进 ... 语句n; } ←缩进</pre>
--	--	--

10、多种语句相互嵌套，则依次缩进



```
if (a==10)
{
    swicth(x)
    {
        case 1:
            a++;
            break;
        case 2:
            a+=2;
            break;
        default:
            ...;
            break;
    } ←不缩进
} ←不缩进
```

```
if (a==10) {
    swicth(x) {
        case 1:
            a++;
            break;
        case 2:
            a+=2;
            break;
        default:
            ...;
            break;
    } ←不缩进
} ←不缩进
```

```
if (a==10) {
    swicth(x) {
        case 1:
            a++;
            break;
        case 2:
            a+=2;
            break;
        default:
            ...;
            break;
    } ←缩进
} ←缩进
```



11、多种语句相互嵌套，则依次缩进(续)

```
switch(x)
{
    case 1:
        for(i=1; i<=100; i++)
            sum+=i;
        break;
    case 2:
        if (b==0)
        {
            b++;
            c--;
        } ← 不缩进
        else
        {
            b++;
            c--;
        } ← 不缩进
        break;
} ← 不缩进
```

```
switch(x) {
    case 1:
        for(i=1; i<=100; i++)
            sum+=i;
        break;
    case 2:
        if (b==0) {
            b++;
            c--;
        } ← 不缩进
        else {
            b++;
            c--;
        } ← 不缩进
        break;
} ← 不缩进
```

```
switch(x) {
    case 1:
        for(i=1; i<=100; i++)
            sum+=i;
        break;
    case 2:
        if (b==0) {
            b++;
            c--;
        } ← 缩进
        else {
            b++;
            c--;
        } ← 缩进
        break;
} ← 缩进
```

12、其它未列举的、更复杂的嵌套组合

★ 以4个空格为单位，按层次依次缩进，每层比上层缩进四格

★ 不符合要求者，每题扣10%的格式分





★ 严重违规（以下分值均为期末总分）

- 打表输出，直接扣除总分20分!!!

举例1：迭代法求2的平方根，直接cout << 1.414;

举例2：1-9组成1953，直接540行输出

- 作业抄袭：所有被发现的，无论抄袭者还是被抄袭者，均不及格

★ 抄袭往届作业，被发现则不及格

★ 找人代做，被发现则不及格

★ 可以挑战，承担后果即可



★ 扣分项及扣分比例（以每题100分计算，自行折算）

- 首行错误 -10分
- 任一位置格式错误 -10分
- 编译报error -100分
（若要求多编译器，则任一编译器报错均-100分）
- 编译报warning -50分/每编译器
（若要求多编译器，则任一编译器报错均-50分）
- 违规使用禁用知识 -100分
例：goto
- 违规使用后续知识 -100分
例：分支部分用for/while；循环部分用函数/数组
- 编译正确，进入测试，按测试点的数量及测试结果扣分
例：20个测试点，3个错误，则成绩为85分

注：每题最多扣到0分为止，不倒扣