

## 第3章 基本程序设计

### 一、判断题

- ( × ) 1. if 语句结构中必须有 else 语句。
- ( × ) 2. 表达式  $7>=3+4 \parallel 6<4 \&\& 2<5$  的求值结果为假。
- ( √ ) 3. 算术运算符的优先级高于比较运算符。
- ( × ) 4. 表达式  $4>6 \parallel 10>2*6 \&\& 2!=5$  求值结果为真。
- ( √ ) 5. 语句 `for(i=0,x=0;i<90;i++) if(i) x++;` 执行完后, x 的值是 89。
- ( × ) 6. 逻辑表达式  $10<x<20$  表示“大于 10 而小于 20 的数”。
- ( √ ) 7. 表达式  $1<3 \&\& 5<7$  的值是 1。
- ( √ ) 8. 表达式  $!(5<8) \parallel 2<6$  的值是 1。
- ( √ ) 9. 表达式 `a+b<c&& d==5` 中运算符优先级由高到低的排列顺序是 +、<、== 和 &&。
- ( √ ) 10. 表达式 `x*y>z&& x*y<100 || x*y>0` 中运算符优先级由高到低的排列顺序是：  
\*、<和>、&&、||。
- ( √ ) 11. 设“`int a=3,b=4,c=5;`”; 表达式“`(a+b)>c&& b==c`”的值是 0。
- ( × ) 12. x,y,z 为 int 类型的时候, 下列语句执行之后, x 的值为 10, y 的值为 20, z 的值为 10。  
`x=10;y=20;z=30;`  
`if(x>y) x=y;y=z;z=x;`
- ( × ) 13. break 和 continue 语句的作用都是终止本层循环的执行。
- ( × ) 14. 判断变量 x 的数值是否在 3 到 5 之间可以用关系表达式  $3<x<5$  来描述。
- ( × ) 15. 判断以下程序是否正确。
- ```
#include <iostream>
#include "iomanip"
using namespace std;
void main()
{   if(a>b)
        cout<<setw(4)<<a;
        cout<< setw(4)<<b;
    else
        cout<< setw(4)<<b;
        cout<< setw(4)<<a;
}
```
- ( × ) 16. 不管是当型循环还是直到型循环, 其循环体都可能在某种条件下一次也不执行。
- ( √ ) 17. 为了避免可能出现的歧义, C++ 对 if...else 语句配对规则规定为: else 总是与最近的没有配对的那个 if 配对。
- ( × ) 18. 要运行一个 C++ 程序需要经过编辑、编译、链接和运行四个阶段。其中编译阶段就是对源程序进行语法与语义分析, 查找和排除程序错误, 通常能够找出程序中的语法错误和逻辑错误。
- ( × ) 19. switch 语句结构中必须有 default 语句。
- ( √ ) 20. break 语句只能结束包含该语句的一层循环结构。

**批注 [11]:** 此为短路表达式,  $7>=3+4$  为真, 则整个表达式为真

**批注 [12]:**  $!$  表示对 5 取非, 5 为非 0 值,  $!$  即为 0, 表达式  $2!=0$  为真, 而  $10>2*6$  为假, 则右边的“与”表达式的值为假, 且  $4>6$  为假, 故整个表达式的值为假

**批注 [13]:**  
`for(i=0,x=0;i<90;i++)`  
`if(i) x++;` //0 为假, 非 0 为真  
除了  $i=0$  时, if 条件为假, 其余 i 值时 if 条件式均为真, 故 x 最后的值为 89

**批注 [14]:** C++ 中正确表达为:  
`X>10&& X<20`

**批注 [15]:** 此为短路表达式,  $2<6$  为真, 则整个表达式的值为真

**批注 [16]:** 此为短路表达式,  $b==c$  为假, 故整个表达式的值为假

**批注 [17]:** 可理解为:  
`x=10;y=20;z=30;`  
`if(x>y)`  
`x=y;`  
`y=z;`  
`z=x;`

**批注 [18]:**  
`if(a>b)`  
`cout<<setw(4)<<a; //if 语句`  
`cout<<setw(4)<<b; //与 if 语句顺序`  
`else`  
`cout<<setw(4)<<b;`  
`else “落单”, 没有和它匹配的 if`

**批注 [19]:** 编译只能发现语法错误

## 二、单项选择题

1. for(i=0, x=0; !x&& i<=3; i++)循环的次数为 ( A )。
- (A) 4 (B) 5 (C) 1 (D) 是无限循环

2. 若有如下程序段：

```
{ int a=2,b=-1,c=2;
  if(a<b)
    if(b<0)c=0;
  else c+=1;
  cout<< "c="<<c<<endl;
}
```

执行后输出到屏幕的结果是\_\_A\_\_。

- (A) 2 (B) 1 (C) 3 (D) 0

3. 若有如下程序段：

```
int i=10;
switch(i)
{ case 9: i++;
  case 10: i++;
  case 11: i++;
  default: i++;
}
```

执行后变量 i 的正确结果是 ( C )。

- (A) 11 (B) 12 (C) 13 (D) 14
4. 以下描述中正确的说法是 ( B )。
- (A) break 语句与 continue 语句完全一样，可以相互代替
- (B) 用 break 语句可以提前终止整个循环
- (C) 用 continue 语句可以提前终止整个循环
- (D) 用 break 语句与 continue 语句都不能提前终止执行循环

5. 以下 for 循环的执行次数是 ( B )。

```
for(x=0,y=0; (y=123)&&(x<2); x++)
```

- (A) 执行 1 次 (B) 执行 2 次 (C) 执行 3 次 (D) 是无限循环
6. 以下描述正确的是 ( C )。

- (A) goto 语句只能用于退出多层循环
- (B) switch 语句中不能出现 break 语句
- (C) 只能用 continue 语句来终止本次循环
- (D) 在循环中 break 语句不能独立出现
7. 为了避免嵌套的 if-else 语句的二义性，C 语言规定 else 总是与 ( C ) 组成配对关系。
- (A) 缩排位置相同的 if (B) 在其之前未配对的 if
- (C) 在其之前未配对的最近的 if (D) 同一行上的 if
8. break 语句的作用是 ( A )。

- (A) 立即跳出包含该 break 语句的最小的各种循环语句和 switch 语句
- (B) 立即跳出包含该 break 语句的最小的各种循环语句
- (C) 立即跳出包含该 break 语句的 switch 语句
- (D) 立即跳出包含该 break 语句的各种循环语句和 switch 语句
9. 下面程序段执行完后，x 的值是 ( C )。

**批注 [110]:** 循环条件解读为：  
x 初值为 0，!x 即为 1（真），此时考察 i<=3 是否成立，则 i 从 0 到 3，共 4 次变化，所以循环次数为 4

**批注 [111]:** else 总是和最近的没有配对的 if 进行配对！

```
if(a<b)
  if(b<0) c=0;
  else c+=1;
cout<<"c"<<c<<endl;
```

**批注 [112]:** 该 switch 语句未加 break，  
执行效果：找到对应开关语句后执行，  
并会顺次执行后面所有的语句！

**批注 [113]:** 循环条件解读为：  
y=123 为赋值语句，即 y 为非 0 值，  
为真，此时考察 x<2,x 从 0 到 1，  
总共变化 2 次，故循环执行的次数为 2

**批注 [114]:** for(i=0,x=0;i<90;i++)  
if(i) x++; //0 为假，非 0 为真  
除了 i=0 时，if 条件为假，其余 i 值  
时 if 条件式均为真，故 x 最后的值为  
89。和判断题第 5 题一样！

- x=0;  
for(i=0;i<90;i++)  
if (i) x++;
- (A) 0 (B) 30 (C) 89 (D) 90
10. continue 语句的作用是 ( C )。
- (A) 结束包含该 continue 语句的最小的各种循环语句  
(B) 结束包含该 continue 语句的各种循环语句  
(C) 结束本次循环, 直接进行循环条件的判断  
(D) 循环执行完后, 结束循环
11. 下列程序段循环 ( A ) 次。
- ```
int x=-10;
while (++x) { }
```
- (A) 9 (B) 10 (C) 11 (D) 无限
12. 在下列运算符中, 优先级最低的是 ( A )。
- (A) || (B) != (C) < (D) +
13. 下列描述正确的是 ( D )。
- (A) 表示  $m > n$  为 false 或  $m < n$  为 true 的表达式为  $(m > n \&\& m < n)$   
(B) switch 语句结构中必须有 default 语句  
(C) if 语句结构中必须有 else 语句  
(D) 如果至少有一个操作数为 true, 则包含 || 运算符的表达式为 true
14. 如果 switch 语句选择表达式中是整型变量, 下面哪一项 case 子句是合法的 ( B )。
- (A) case "2": (B) case 2: (C) case 2; (D) case ==2
15. 有如下程序段:
- ```
.....
for(int i = 1; i < 4; i++)
{ if(i%2 == 0) continue;
  cout << i << " "; }
```
- 执行以上程序段的输出结果是 ( C )。
- (A) 1, (B) 1,3,4, (C) 1,3, (D) 2,4,
16. 下列运算符中优先级最高的是 ( A )。
- (A) ! (B) % (C) -= (D) &&
17. 用逻辑表达式表示 “大于 10 而小于 20 的数”, 正确的是 ( D )。
- (A)  $10 < x < 20$  (B)  $x > 10 || x < 20$  (C)  $x > 10 \&\& x < 20$  (D)  $!(x \leq 10 || x \geq 20)$
18. 如果 switch 语句中的选择表达式是字符型变量 code, 下面哪一项 case 子句是合法的 ( B )。
- (A) case "3"; (B) case '3'; (C) case 3; (D) case =3
19. 设  $\text{int } a=10, b=11, c=12; (a+b) < c \&\& b==c$  的值是 ( B )。
- (A) 2 (B) 0 (C) -2 (D) 1
20. 在以下关于 C++ 语言的叙述中, 正确的是 ( D )。
- (A) if 语句结构中必须有 else 语句 (B) switch 语句结构中必须有 default 语句  
(C) C++ 程序必须有 return 语句 (D) C++ 中使用流来执行标准的输入输出操作
21. 下列 do-while 循环的循环次数是 ( D )。
- 已知:  $\text{int } i=5;$   
do{ cout << i-- << endl;

**批注 [115]:** 答案是 A。

你需要理解 while(++X) 的意思, 对于前置自增运算, 是先对 x 的值加 1 作为整个表达式的值, 即第一次 ++X=9, 对于 while 的循环条件而言, 只要是非零值均为真, 故会进行一次循环操作, 直到 ++x 的值为 0, 即从 -9 到 -1, 一共完成了 9 次循环操作!

等价于如下代码:

```
x=-10;
x=x+1; //前置运算, 先加 1
while(x!=0)
{
    ....;
    x++;
}
```

**批注 [116]:** continue 的作用是结束本次循环! 从题目看, 如果 i 是偶数则不会输出, 注意 i 的取值是从 1~3, 故最后输出就是 1,3,

**批注 [117]:** C 少了一个 & 符号, 如果只有一个 & 表示是二进制的 “与” 运算! 答案 D 是对的, 它是对 “小于等于 10 或大于等于 20” 取反 (!), 故结果就是 “大于 10 且小于 20”

```
i--;  
}while (i!=0);
```

(A) 0 (B) 2 (C) 5 (D) 无限次

22. 若 m 是一个值为 10 的 int 型变量, n 是一个 bool 型变量, 则表达式 !m||n>0 的值( D )。

(A) 为 true (B) 为 false (C) 与 n 的值相反 (D) 与 n 的值相同

**批注 [118]:** i 在循环中需要减两次, 则 i 对于 while 的条件而言, 始终取不到 0 值, 所以非零值均为真, 故循环会一直做!

**批注 [119]:** m 初值为 10, 则 !m 为 0, 而 !m||n>0 是一个“或”连接的短路表达式, 所以, 整个表达式的值就由 n>0 决定, 又 n 为 bool 值, 其值为真 (1) 或假 (0), 故整个表达式的值为值与 n 的值相同!