

第二部分：答案与详细解析

1. 答案：C

- 解析：
 - A 错误：标识符不能以数字开头。
 - B 错误：`int` 是C语言的关键字（保留字），不能用作变量名。
 - C 正确：标识符只能由字母、数字和下划线组成，且以下划线或字母开头。
 - D 错误：减号 `-` 不允许出现在标识符中。

2. 答案：C

- 解析：这是初学者最容易犯的错。
 - `if (i = 0)` 中使用的是赋值运算符 `=`，而不是比较运算符 `==`。
 - 表达式 `i = 0` 的值为 0。
 - 在C语言中，0 代表“假”，非 0 代表“真”。
 - 因为条件为假，所以执行 `else` 分支，`k` 被赋值为 2。

3. 答案：B

- 解析：C语言中基本数据类型的函数传参是**值传递**（Pass by Value）。这意味着系统会开辟新的内存空间（形参），并将实参的值复制进去。对形参的修改不会影响实参。

4. 答案：A, B, C

- 解析：
 - A 正确：`while` 是当型循环，先判断。
 - B 正确：`do-while` 是直到型循环，先执行一次再判断。
 - C 正确：`do-while` 结构尾部必须有分号。
 - D 错误：如果是 `do-while`，即使条件一开始为假，循环体也会执行一次。

5. 答案：C

- 解析：
 - A 错误：`a` 是实参（调用时传入），`x` 是形参（定义时声明）。
 - B 错误：因为是值传递，`change` 函数里的 `x` 只是 `a` 的一份拷贝。
 - C 正确：`x` 是局部变量，函数结束即销毁，`a` 不受影响。
 - D 错误：它们占用不同的内存地址。

6. 答案：B

- 解析：
 - 注意运算优先级和类型转换顺序。
 - `a` 和 `b` 都是整数，`a / b` 执行的是**整数除法**，即 `5 / 2 = 2`（截断小数部分）。
 - 然后再将整数 `2` 强制转换为 `float`，结果是 `2.0`。
 - 如果想要得到 2.5，应该写成 `(float)a / b`。

7. 答案：A, B

- 解析：

- A 正确：这是著名的“悬挂 else”原则。
- B 正确：花括号 `{}` 定义了代码块，可以强制改变作用域和配对关系。
- C 错误：C语言不以缩进决定逻辑结构（Python才是），缩进只为了可读性。
- D 错误：`if` 可以独立存在，不需要非得配 `else`。

8. 答案：B

- 解析：

- 第一次：i=0，条件 0<5 (真)，输出，i变为2。
- 第二次：i=2，条件 2<5 (真)，输出，i变为4。
- 第三次：i=4，条件 4<5 (真)，输出，i变为6。
- 第四次：i=6，条件 6<5 (假)，循环结束。
- 共执行3次。

9. 答案：A, C

- 解析：

- A 正确：函数内定义的为局部变量。
- B 错误：局部变量随函数结束而销毁；全局变量或 `static` 变量才贯穿整个程序。
- C 正确：这叫“就近原则”或“局部优先”。
- D 错误：局部变量如果不初始化，其值是随机的（垃圾值）；全局变量默认初始化为0。

10. 答案：B

- 解析：

- 这是逻辑与 `&&` 的**短路特性**。
- 表达式为 `A && B`。
- 首先计算 A：`a > 5` 即 `1 > 5`，结果为假 (0)。
- 对于 `&&` 运算，只要左边为假，整个表达式一定为假。编译器为了优化，**不会再执行右边的表达式**。
- 因此，`b++` 根本没有被执行。
- 最终 `a` 还是 1，`b` 还是 1。