C++ while 循环 课后作业(二)

姓名: _____

一、 填空题

1. 在编程中，我们用来存放滚雪球式增长的和的变量，通常被称为 _____，它的初始值在进行加法运算时通常设为 _____。
2. `sum = sum + i;` 这行累加代码可以简写为 _____。
3. 在 C++ 中，数值 _____ 被视为“假”（false），而所有 _____ 的数值都被视为“真”（true）。
4. 使用 `while` 循环挨个访问某个区间内的每一个元素，并在循环内部用 `if` 语句挑出符合特定条件的元素，这种编程模式被称为_____。
5. 想要实现“循环n次”的效果，那么当循环控制变量 `i` 从 `0` 开始时，`while()` 中的循环条件则应该写为 _____。

二、 单项选择题

(　) 1. 分析以下代码，最终输出的结果是什么?

```
int sum = 0, i = 1;
while (i <= 4) {
    sum = sum + i;
    i++;
}
cout << sum << endl;
```

A. 4 B. 5 C. 10　　D. 15

(　) 2. 如果将上一题中的 `cout << sum << endl;` 语句放到 `while` 循环体的内部，程序的输出会变成?
A. 只输出 10　　B. 什么都不输出　　C. 依次换行输出 1, 3, 6, 10　　D. 程序会报错

(　) 3. 要打印出 1 到 100 之间所有“个位数是7”的数字，循环体内部的 `if` 判断条件应该写成?
A. `if (i == 7)`　　B. `if (i % 7 == 0)`　　C. `if (i / 10 == 7)`　　D. `if (i % 10 == 7)`

(　) 4. 运行 `int n=5; int i=0; while(i<n){ cout << "*"; i++; }` 这段代码，会输出多少个 `*`?
A. 0 个　　B. 4 个　　C. 5 个　　D. 6 个

(　) 5. 关于 `while(-1)` 这个循环条件，下列说法正确的是?
A. 条件为假，循环一次都不会执行
B. 条件为真，会形成死循环
C. 程序会因为负数而出错
D. 循环只会执行一次

(　) 6. 以下代码段的目的是计算 1 到 10 之间所有偶数的和，最终 `sum` 的值是?

```
int sum = 0;
int i = 1;
while (i <= 10) {
    if (i % 2 == 0) { sum = sum + i; }
    i++;
}
```

A. 55　B. 30　C. 25　D. 20

(　) 7. 在“遍历+筛选”模式中，`if` 语句扮演的角色是?

A. 确保循环能够停止　　B. 驱动循环变量从起点到达终点

C. 定义筛选规则，挑出所需要的数据　　D. 初始化累加器

(　) 8. 观察下面的代码，当循环结束后 `count` 的值是多少?

```
int count = 10;
while (count > 5) { count--; }
```

A. 10　B. 6　C. 5　D. 4

(　) 9. 考虑以下代码，它的问题是什么?

```
int i = 1;
while (i <= 100) {
    if (i % 2 == 0) {
        cout << i << endl;
        i++;
    }
}
```

A. 这是一个死循环　　B. 它会漏掉一些偶数

C. 它只能打印出奇数　　D. 它在遇到奇数时会陷入死循环

(　) 10. 要计算 1 到 100 以内所有 3 的倍数之和，下面代码片段的 `???` 处应填写?

```
int sum = 0;
int i = 1;
while (i <= 100) {
    if (???) {  sum += i;  }
    i++;
}
```

A. `i / 3 == 0`　B. `i % 3 == 0`

C. `i % 3 != 0`　D. `i * 3 == 0`

三、 判断题 (对的打“√”，错的打“×”)

(　) 1. `while(x)` 和 `while(x != 0)` 这两个循环条件是等价的。

(　) 2. 在累加求和的程序中，如果累加器变量 `sum` 没有被初始化为 0，程序最终的计算结果一定是错误的。

(　) 3. 要筛选出 1 到 100 之间能同时被 2 和 3 整除的数，`if` 条件应写为 `if (i % 2 == 0 || i % 3 == 0)`。

(　) 4. 执行代码 `int i = 0; while(i < 5) { cout << i; i++; }` 会在屏幕上打印出 `01234`。

(　) 5. `while(1)` 会创建一个无限循环，是因为 C++ 认为 `1` 是一个“真”值。

(　) 6. 在“遍历+筛选”模式中，`i++` 必须放在 `if` 语句的外面（但在循环体内），这样才能保证无论当前数字是否被选中，都能继续检查下一个数字。

(　) 7. `while` 循环只适用于我们提前知道循环次数的场景。

(　) 8. `sum += i;` 的执行效率比 `sum = sum + i;` 更高。

(　) 9. 初始化累加器 `sum` 为 0 的主要目的是为了让代码能够通过编译。

(　) 10. `i = 0; while(i < n)` 和 `i = 1; while(i <= n)` 都能实现循环 n 次的效果。