C++ switch 多选择语句 课后习题

姓名: _____

一、 填空题

1. `if-else if` 的逻辑是“顺序比较”，而 `switch` 的逻辑是“_____”，在选项多时效率更高。
2. `switch` 语句中，每个选项都由关键字 _____ 开头，后面跟一个常量值和冒号。
3. 为了防止“case 穿透”，在每个独立的 `case` 代码块结束时，都应该加上 _____ 语句。
4. 如果 `switch` 表达式的值与所有 `case` 都不匹配，程序会执行 _____ 分支下的代码。
5. 一个 `case` 后面忘记写 `break`，程序会继续向下执行之后的所有代码，这种现象叫做“_____”。

二、 单项选择题

( ) 1. 阅读代码，当输入为 `2` 时，输出结果是什么？

```
int n;
cin >> n;
switch(n) {
    case 1: cout << "Apple"; break;
    case 2: cout << "Banana"; break;
    case 3: cout << "Cherry"; break;
}
```

A. Apple　B. Banana　C. Cherry　D. 没有输出

( ) 2. 阅读代码，当输入为 `2` 时，输出结果是什么？（注意：缺少 `break`）

```
int n;
cin >> n;
switch(n) {
    case 1: cout << "A";
    case 2: cout << "B";
    case 3: cout << "C";
}
```

A. B　B. BC　C. ABC　D. C

( ) 3. 阅读代码，当输入为 `5` 时，输出结果是什么？

```
int n;
cin >> n;
switch(n) {
    case 1: cout << "A"; break;
    case 2: cout << "B"; break;
    default: cout << "Other";
}
```

A. A　B. B　C. Other　D. 没有输出

( ) 4. `switch` 语句主要用来替代哪种结构的 `if` 语句?

A. `if (score > 90)`

B. `if (a > 0 && b > 0)`

C. `if (day == 1) { ... } else if (day == 2) { ... } else if (day == 3) { ... }`

D. `if (password != "123")`

( ) 5. 下列哪种判断**不能**直接用 `switch` 语句实现?

A. 判断某天是周几　　B. 判断成绩是否及格

C. 判断输入的字符是不是 'Y'　　D. 判断用户的选择是不是 1

( ) 6. `switch (n)` 中，`n` 的结果**不能**是什么类型?

A. `int`　B. `char`　C. `double`　D. `bool`

( ) 7. 在 `switch` 结构中，`default` 关键字的作用是?

A. 提供一个默认的变量值　　B. 当所有 `case` 都不匹配时执行

C. 必须写在 `switch` 的最开始　　D. 用来结束 `switch` 语句

( ) 8. 为了判断一个字符 `c` 是不是元音字母（a, e, i, o, u），最高效的 `switch` 写法是?

A. 为 a, e, i, o, u 分别写五个独立的 `case`，每个都带 `break`

B. 使用 `if (c=='a' || c=='e' ...)`

C. 将 `case 'a': case 'e': case 'i': case 'o': case 'u':` 堆叠在一起，共享一个代码块

D. `switch` 无法实现此功能

( ) 9. 阅读代码，当输入为 `'B'` 时，输出结果是什么?

```
char c;
cin >> c;
switch(c) {
    case 'A': cout << "Apple";
    case 'B': cout << "Banana";
    case 'C': cout << "Cherry"; break;
    case 'D': cout << "Durian";
}
```

A. Banana　B. BananaCherry　C. BananaCherryDurian　D. Cherry

( ) 10. `case` 标签后面跟的值必须是?

A. 一个变量　B. 一个常量　C. 一个表达式　D. 一个函数

三、 判断题 (对的打“√”，错的打“×”)

( ) 1. `switch` 语句只能用于进行“等于”的判断。

( ) 2. 在 `switch` 语句中，`default` 分支是必须存在的，不能省略。

( ) 3. `case` 后面跟的可以是 `case x:`，其中 `x` 是一个 `int` 类型的变量。

( ) 4. `break` 语句的作用是跳出当前的 `switch` 结构。

( ) 5. 如果好几个 `case` 需要执行的操作完全一样，可以利用“case穿透”来简化代码。

( ) 6. `if-else if` 语句能实现的功能，`switch` 语句也一定能实现。

( ) 7. `switch` 语句中，各个 `case` 标签的顺序不影响最终的逻辑结果（假设都正确使用了 `break`）。

( ) 8. 忘记在 `case` 中写 `break` 会导致编译错误（CE）。

( ) 9. `switch` 的逻辑是“直接跳转”，而 `if-else if` 是“顺序比较”。

( ) 10. 可以在一个 `switch` 结构中，放置另一个 `switch` 结构。