

附件 2：试题形式及样题

试题形式

比赛题目完全为客观题型。

➤ 结果填空题

- (1) 题目描述一个具有确定解的问题。要求选手对问题的解填空。
- (2) 不要求解题过程，不限制解题手段，只要求填写最终的结果。

➤ 代码填空题

- (1) 题目描述一个具有确定解的问题。
- (2) 题目同时给出该问题的某一解法的代码，但其中有缺失部分。

要求选手读懂代码逻辑，对其中的空缺部分补充代码，使整段代码完整。只填写空缺部分，不要填写完整句子。不要写注释、说明或其它题目中未要求的内容。

➤ 编程大题

- (1) 题目为若干具有一定难度梯度、分值不等的编程题目。
- (2) 这些题目的要求明确、答案客观。
- (3) 题目一般要用到标准输入和输出。要求选手通过编程，对给定的标准输入求解，并通过标准输出，按题目要求的格式输出解。题目一般会给出示例数据。
- (4) 一般题目的难度主要集中于对算法的设计和逻辑的组织上。理论上，选手不可能通过猜测或其它非编程的手段获得问题的解。
- (5) 选手给出的解法应具有普遍性，不能只适用于题目的示例数据（当然，至少应该适用于题目的示例数据）。
- (6) 为了测试选手给出解法的性能，评分时用的测试用例可能包含大数据量的压力测试用例，选手选择算法时要尽可能考虑可行性的问题。

样题

【编程大题】信用卡号验证

当你输入信用卡号码的时候，有没有担心输错了而造成损失呢？其实可以不必这么担心，因为并不是一个随便的信用卡号码都是合法的，它必须通过 Luhn 算法来验证通过。

该校验的过程：

- 1、从卡号最后一位数字开始，逆向将奇数位(1、3、5 等等)相加。
- 2、从卡号最后一位数字开始，逆向将偶数位数字，先乘以 2（如果乘积为两位数，则将其减去 9），再求和。

3、将奇数位总和加上偶数位总和，结果应该可以被 10 整除。

例如，卡号是：5432123456788881

则奇数、偶数位（用红色标出）分布：5432123456788881

奇数位和=35

偶数位乘以 2（有些要减去 9）的结果：1 6 2 6 1 5 7 7，求和=35。

最后 35+35=70 可以被 10 整除，认定校验通过。

请编写一个程序，从标准输入获得卡号，然后判断是否校验通过。通过显示：“成功”，否则显示“失败”。

比如，用户输入：356827027232780

程序输出：成功

【程序测试参考用例】

356406010024817 成功

358973017867744 成功

356827027232781 失败

306406010024817 失败

358973017867754 失败

【c/c++组代码填空】

下列代码把一个串 p 复制到新的位置 q。请填写缺少的语句：

```
char* p = "abcde";  
char* q = (char*)malloc(strlen(p)+1);  
for(int i=0;_____; i++) q[i] = p[i];  
q[i] = 0;
```

【参考答案】

p[i] 或 *(p+i) 或 p[i] != '\0'

（答案不唯一，以选手提供代码带入专用验证程序测试为依据，验证程序比题面中提供的程序片段更严谨，更完善）