

# Chapter 1

## 编程类练习参考评分标准

### 1.1 编程类练习评分标准 (有报告)

#### 1.1.1 设计考虑

通过此类练习希望学生能够通过应用掌握一些工程工具，如此课程中的 git 和流程图的绘制软件及绘制方法。

另外希望学生能够进一步熟练使用 C 语言进行编程，提高使用 C 语言的能力。

学生能够理解已有的密码算法，并能够用 C 语言实现。

学生能够利用 GMP 大数据库编写算法，具备初步利用第三方库的能力，同时可以利用他设计并实现一些底层的数学算法。

通过编写规范的技术文档，基本掌握此类文档的编制方法。

#### 1.1.2 参考评分标准

每次代码评分总分，如果没有按时提交为 0 分，没有按时提交，但是有合理理由，并被老师接受，为了体现公平，其评分总分为此次按时交作业中的最低分，最后得分将按其实际总分等比例获得。

每次代码评分总分为 60 分，评分规则为：

- CR(code running): 代码运行情况，共 35 分，细则为：
  - 无提交 0 分.
  - 能按要求正确运行并且在验证执行中没有错误，35——30 分；
  - 能按要求正确运行但在验证执行中有一些错误，不影响获得正确结果，30——25 分；
  - 基本能够按照要求运行，运行中有错误，但可以获得正确的主要结果，25——20 分.
  - 可以运行，无法获得正确结果，20——15 分.
  - 有可执行程序，但无法运行，获得合理解释，15——0 分.
- DQ(document qualification): 文档文字描述，共 15 分，细则为：

- DQ-1 文档中有文字描述. 无 0 分, 有 5 分.
- DQ-2 文字描述质量. 简练清晰 5 分, 把框架或功能准确描述出来 3, 描述混乱 0.
- DQ-3 有框架或功能流程图. 无 0 分, 不满足规范或者不准确 3 分, 满足规范且准确 5 分。
- TC(test cases): 测试用例, 共 10 分, 细则为:
  - TC-1 测试用例. 无 0 分, 有但是描述不规范 3 分, 有且描述规范 5 分.
  - TC-2 测试用例设计. 覆盖全面功能且考虑全面 5 分, 覆盖全部功能但考虑不全面主要内容 3, 未覆盖全面功能且考虑欠缺 0 分。

## 1.2 编程类练习评分标准 (无报告)

此类练习只需要提交源代和编译后的代码到 gitee 上即可。

评价依据是, 如果没有按时提交为 0 分, 没有按时提交, 但是有合理理由, 并被老师接受, 为了体现公平, 其评分总分为此次按时交作业中的最低分, 最后得分将按其实际总分等比例获得。

代码评分总分为 50 分, 评分规则为:

- CR(code running): 代码运行情况, 共 35 分, 细则为:
  - 无提交 0 分.
  - 能按要求正确运行并且在验证执行中没有错误, 35——30 分;
  - 能按要求正确运行但在验证执行中有一些错误, 不影响获得正确结果, 30——25 分;
  - 基本能够按照要求运行, 运行中有错误, 但可以获得正确的主要结果, 25——20 分.
  - 可以运行, 无法获得正确结果, 20——15 分.
  - 有可执行程序, 但无法运行, 获得合理解释, 15——0 分.
- EN(explanatory note), 共 15 分, 细则为:
  - EN-1 无注释. 无 0 分, 有 5 分.
  - EN-2 通过注释能很好阅读代码, 注释简练清晰全面 10-15 分; 通过注释不能很好阅读代码 (比如注释不全面等问题), 5-10 分