

## 实验 2: 复习与拓展 2-字符串学习部分

### 类与对象初步

姓名\_\_\_\_\_班级\_\_\_\_\_学号\_\_\_\_\_

- 请阅读此说明：实验 2 满分 100 分，附加分 20 分；做完实验后请按要求将代码和截图贴入该文档。然后将此文档、源代码文件（.cpp）打包上传到学习通。

#### 1、字符串-基础题：（10 分）

编写函数，完成和实现拷贝字符串 n 个字符的功能。

函数原型为： `bool mystrncpy(char *to, char *from, unsigned startpos, unsigned len);`

表示需要从 from 串中的 startpos 位置开始拷贝 len 个字符串内容到 to 串中，返回拷贝是否成功的信息(失败情况如位置无效，长度无效等)。

- **实验要求：**
- ① 按照给定的函数原型定义并实现函数，并且设计主函数进行函数功能的测试。
- **实验提交：**

将完整的源代码和测试截图 粘贴在下面。

- **测试参考：假设 from 串内容为：abcdefghijklmn**

测试时使用可以参考如下可能的调用样本：

```
mystrncpy(to, from, 0,0);  
mystrncpy(to, from, 3,4);  
mystrncpy(to, from, 3,20);
```

■ **源代码粘贴处：**

■ **程序测试截图：**

#### 2、字符串-进阶题：（20 分+ 5 分）

从键盘接受一篇英文文章，统计其包含大写字母、小写字母、数字字符、空格、其他字符的个数，并统计文章的单词数（单词使用空格分隔。）

- **实验要求：**
- ① 按照需求完成程序的设计和实现。
- **实验提交：**

将完整的源代码和测试截图 粘贴在下面。

- **可选-实验拓展：**考虑程序功能拓展到处理多篇英文文章。（附加 5 分），请将设计完成的程序粘贴在后面。

■ **源代码粘贴处：**

■ **程序测试截图：**

■ **拓展源代码粘贴处（可选）：**

### **3、字符串-高阶题：（20 分+10 分）**

使用字符串/数组实现超级整数的存储和基本计算，比如完成一个表示范围可能达 30000 位的超级整数。要求实现其加法、减法、乘法运算。每一种运算定义成函数，并编写主函数测试你的超级整数。自行设计输入输出格式。比如，输入形式为：“数字 1,数字 2,运算符” 例 ” 123456788,3456789012345,+”使逗号分隔，需要判断表达式的合法性，并计算结果。

要求：，并讨论：1、选择使用数组或者使用字符串的理由；2、考虑基本存储类型的选择：char 或者 int？3、讨论你的采用的策略能表示超级整数的极限（可以到多大）

- **实验要求：**

① 按照需求完成程序的基本功能设计和实现。

**可选-**设计并实现两种方式的输入处理：计算需要的大数可以由随机数生成。小数可以提供用户输入测试（基本功能）。（附加 5 分）

② **可选-**思考：1、考虑基本存储类型的选择：char 或者 int？选择使用数组或者使用字符串的理由；2、讨论你的采用的策略能表示超级整数的极限（可以到多大）（附加 5 分）

- **实验提交：**

将完整的源代码和测试截图 粘贴在下面。

■ **源代码黏贴处：**

■ **程序测试截图：**

■ **输入设计拓展（可选）：**

■ **思考（可选）：**

### **4、类与对象初步-基础题：（10 分）**

根据自己的掌握情况重做课堂例题：游泳池造价计算 + 停车收费系统

- 注意练习和熟悉多文件装配模型:

一个工程项目包含: 自定义类库 `xxx.hpp+xxx.cpp` 以及测试的主程序 `maintest.cpp`

如 游泳池造价计算 包含 `pool.hpp , pool.cpp , pooltest.cpp`

停车场收费系统 包含 `time.hpp , time.cpp , ParkingCost.cpp`

### 5、停车场计费系统的升级思考: (40分+5分)

以课堂材料给出的停车场系统为基础考虑(见学习通课件材料), 如果 停车收费规则改为:

- 1小时内免费;

- 1小时以上:

超过 15 分钟不到 30 分钟按半小时收费;

超过 30 分钟不到 1 小时按 1 小时收费;

超过 1 小时按每小时 4 元收费;

- 超过 1 天, 每天按 30 元收费; 停车超过 1 个月的, 停车费打 9 折;  
超过 1 年的停车费打 8 折。

- 实验要求:

① 考虑修改 `Time` 的设计, 改为 `DateTime`, 补充数据成员年、月、日, 补充或者调整类内成员函数的设计, 并实现它。采用你实现的新的 `DateTime` 进行停车收费的测试。并将完整的源代码和测试截图黏贴在下面。(20分)

源代码只需要黏贴类定义部分即可 (DateTime 的声明和实现)

② 考虑另一种程序调整模型, 补充 `Date` 类, 数据成员部分为: 年、月、日; 成员函数部分可以仿照原来 `Time` 类的功能设计。采用补充的新类 `Date` 和原来的 `Time` 类一起工作完成停车收费程序。(20分)

\*并思考: 在新的收费规则下, 两种程序调整的策略差异导致调整的工作量不同, 你更倾向哪一种? 给出你的理由 (附加 5 分)

①

■ 源代码粘贴处:

■ 程序测试截图:

②

■ 源代码粘贴处:

■ 程序测试截图:

\*附加思考: