“模拟外卖派单”大作业TEAM28编码规范

一、文件体系结构

1. c文件都在文件头加入文件注释：

/\*@file: @author: @version: @date：\*/

2. H头文件结构：

#ifndef \_DATA\_DEF\_H\_

#define \_DATA\_DEF\_H\_

#inlcude <stdio.h>

/\*include区\*/

#define MM 30

/\* 宏定义区 \*/

/\* 结构、类、变量等定义区 \*/

/\* 输出函数定义区 \*/

extern int test(void);

#endif

3.源文件包含：

main.c//主实现文件

input.c//输入和订单存储模块

graph.c// 建图模块

monitor.c//监控、更新模块

output.c//输出模块

test.c// 测试模块

4.工程中不起作用的文件应删除，工程目录下的非工程文件也应该移走，保持工程的清洁，避免混淆难于管理。

二、排版规则

1.在每个函数定义结束之后都要加空行。

2.程序排版使用缩进

3.代码行不要过长，最大长度宜控制在70至80个字符以内，长表达式要拆分成新行，拆分出的新行要行缩进

4.修改代码中要加入修改内容、修改人、修改时间

5.不再有用的注释要删除

三、变量和结构

主要变量包含：

int start[][];//骑手的起点位置

int pro;//收益

int t;//时间

int riderNum;//所需骑手数量

主要结构包含：

struct orderList//订单信息链表

{

int num\_order;//订单编号

int x\_res,y\_res;//餐馆位置

int x\_cli,y\_cli;//食客位置

int dist;//餐馆与食客之间的路径长度

int t0[];//订单开始派单时刻

struct orderlist \*nextOrder;

}

struct orderTime//监控订单时间情况

{

int t\_past[];//订单已过时间

int t\_left[];//订单剩余时间

int t\_end[];//订单结束时刻

int t\_over[];//订单超出时间

}

struct rider//骑手状态

{

int num\_rider[];//骑手编号

int x\_rid,y\_rid;//骑手位置

int dist\_res[][],dist\_cli[][];//骑手与餐馆或食客的距离

int order[][];//所接单序号（包括已接的单和即将接的单）

int order\_now[];//骑手正在派的单的序号

}

struct graph//图的存储

{

int arc[][];//邻接矩阵

int tag[][];//在目标点上标记餐馆（1）、食客（0）、无（-1）

int vis\_res[][],vis\_cli[][];//标记访问过的餐馆与食客

}

1. 变量的定义尽可能放在函数最开始处，不要在函数体的中间位置随意定义变量。在定义变量的同时初始化该变量

2.需要对外公开的常量放在头文件中，不需要对外公开的常量放在定义文件的头部

3. i++，禁止嵌入其他表达式，只能单独一行，唯一的例外是for结构中的递增。

四、函数和过程

1.函数设计时必须把函数原型放在main函数之前，而函数实现放在main函数之后

2.明确函数的功能，不要把所有的功能都放到一个函数体内执行

五、质量保证/安全规则

1.系统应具有一定的容错能力，对一些错误事件（如用户误操作等）能进行自动补救。

2.过程/函数中分配的内存，在过程/函数退出之前要释放