



# 棉花模拟采集系统

# 终期报告

c 语言课程设计



专业班级: 自动化 2202

小组成员: 冯天瑞(U202214953) 梁栢杰(U202290032)

指导老师: 周纯杰、何顶新、汪国有、左峥嵘

周凯波、彭刚、高常鑫、陈忠

上交时间: 2023年4月22日

# 目录

<b>-</b> 、	编写背景	3
二、	目标功能	4
三、	运行环境与配置	4
四、	需求分析	5
五、	系统设计	6
六、	界面设计	12
七、	函数代码	23
八、	时间安排	142
九、	工作分配及代码量	143
十、	组员感想	143
+-	−、参考资料	144

## 一、编写背景

21 世纪初,互联网飞速发展并迅速普及到各行各业,而属于基层的农作物业也不再靠传统的纯人力耕种。收割机,运输车,智能仓库等近现代的工业机械也逐渐走进了中国的田地。

而棉花行业就是农业中最重要的产业之一,它产量大,生产成本低,使棉织品价格比较低廉。棉花能制造成多种规格的纺织物,衣服,布,棉签,手套,鞋子,口罩,被子,甚至钞票和医用绷带等都可以由棉花制造而成,可谓用途广泛,在我们的生活中随处可见了。棉花从三千多年前就已经传入中国,但一直到元朝时期才开始广泛种植,多数种植在西北地区,黄河、长江流域。

棉花在生活中的应用已非常广泛,棉花中的棉可以卖钱还能做成棉被,棉衣在冬季成为生活取暖的重要物品,而且棉籽还可以用来榨油满足生活的需要,剩下的秸秆还可以作为燃料生火取暖和做饭,棉花的皮还可以磨成肥料饲养牲畜。

棉花的种植一般分为播种,管理和采集三个部分,而本项目主要体现在采集和管理的部分。通过棉花种植园的面积和地理位置,计算棉花的收获方式和收获量,模拟棉花采集的全过程。

# 二、目标功能

本项目主要通过模拟棉花采集的过程。根据其投入的收割机数量和型号、种植棉花的地理位置、棉花田的面积和形状,计算出其产出的棉花量、收割时间和收割形式,并模拟和制作出棉花从采集到装库全过程的动画。以此提供用户对自己实际种植棉花情况的参考和建议收获方式。该项目通过鼠标与键盘直接进行控制。用户将鼠标移至需要操作的区域进行点击来显示不同界面,同时通过键盘来完成各种参数的输入功能。

# 三、运行环境和配置

#### 一、硬件接口

处理器: Intel Pentium 166 MX 或以上。

硬盘: 空间 500MB 以上。

屏幕适配器: VGA 接口。

系统运行内存: 要求 32MB 以上。

#### 二、软件接口

开发软件工具: Borland C++

文字编辑工具: visual Stdio Code

操作系统: DOS WINDOWS 9X/ME/2000/XP/WINDOWS 10/WINDOWS 11

# 四、需求分析

#### 棉花自动采摘系统模拟

主要功能说明:根据农田面积进行采摘路径规划、模拟智能农机 进行采摘作业、对采摘棉花进行统计、并运输到棉花集中站进行储存、 盘点、出库和入库等功能。

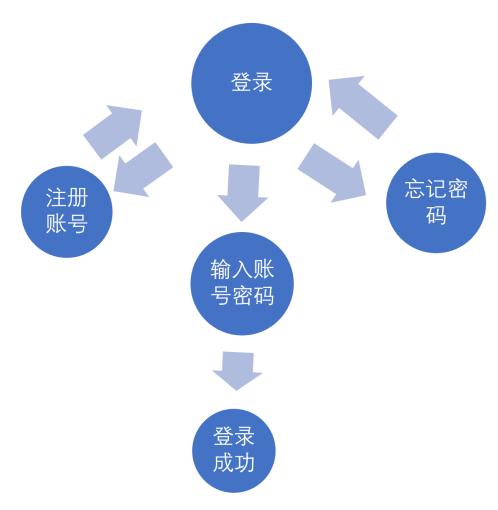
根据本选题要求进行需求分析,可见要求制作的软件系统是一款棉花采摘农业实践的仿真模拟系统,需要最大程度地结合实际,为实际农业自动化生产服务,现根据题目要求及实际查阅资料,有以下核心用户需求,即软件核心功能:

- 1) 本软件分用户使用,用户登录进入操作主页面,便可开始模拟 操作
- 2) 通过实际情况,用户可以根据不同地区需求设置棉花生产参数, 土地参数,系统自动推荐棉花种植种类,农机类型
- 3) 进入模拟流程,系统根据土地类型,农机类型,自动规划采摘路径,模拟智能农机进行采摘,给出采摘用时。
- 4) 进入仓储界面,自行操作仓库出入库流程
- 5) 进入参数列表,修改过往参数

# 五、系统设计

#### 登录界面流程:

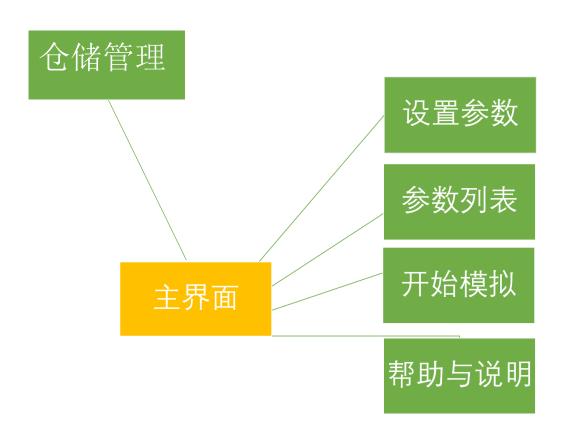
1)输入账号和密码完成登录并进入主界面,也可选"忘记密码"或注册新账号。忘记密码的验证通过注册账号时输入的手机号来确认。注册账号或者找回密码回就会返回登录界面。



#### 主界面流程:

主界面中主要给用户进行选择。用户的功能主要分为:

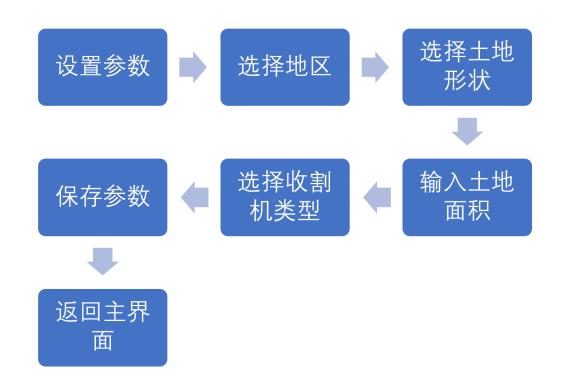
- 1) 设置参数
- 2)参数列表
- 3) 开始模拟
- 4) 仓储管理
- 5)帮助与说明



设置参数流程:

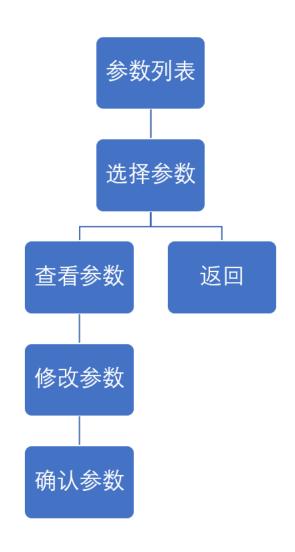
需要设置的参数包括:

- 1)选择地区并输入参数名,本项目提供中国三个适合种植棉花的地区进行选择,包括西北地区,黄河地区和长江地区。
- 2)选择土地形状,本项目提供了三种土地形状,包括矩形, 圆形和三角形。
- **3**)输入土地面积,利用键盘输入在该形状下的土地大致面积。
- 4)选择收割机类型,采集棉花常用的收割机类型分为垂直式和水平式,本项目亦提供这两种选择。
- 5)保存参数并返回主界面。



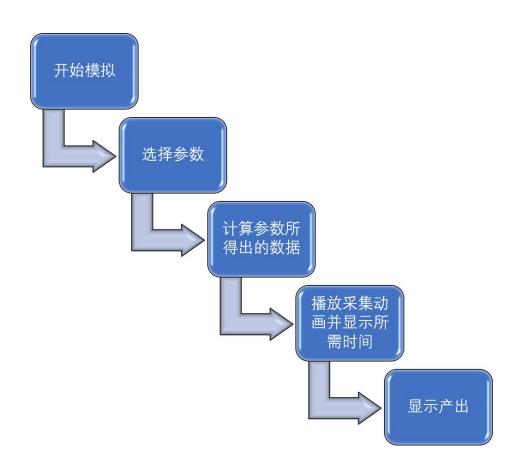
#### 参数列表流程:

- 1) 选择参数,选择之前保存过的参数。
- 2) 查看参数, 查看该参数具体内容。
- 3)修改参数,选择需要修改的参数内容,并进行修改。



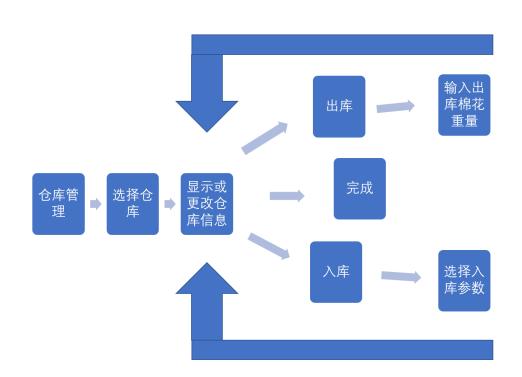
#### 开始模拟流程:

- 1) 根据已编辑的参数选择参数。
- **2**) 计算参数所得出的数据,根据选择的参数,计算出采集 所需时间,采集方式和产出。
- 3)播放采集动画,根据土地面积和土地形状,播放相应的 采集动画。
- 4)显示产出,显示计算出的棉花产出。



#### 仓储管理流程:

- 1) 选择仓库名,或者新建仓库。
- 2) 盘点并显示该仓库的棉花总数,所剩仓库容量。并可进 行出库看或入库的操作改变库存。
- 3) 完成操作后退出仓储管理,并返回主界面。



# 六、界面设计

## 登录界面:

输入账号和密码完成登录并进入主界面,也可选"忘记密码" 或注册新账号。忘记密码的验证通过注册账号时输入的手机 号来确认,验证成功则会提供其令其重新输入密码。

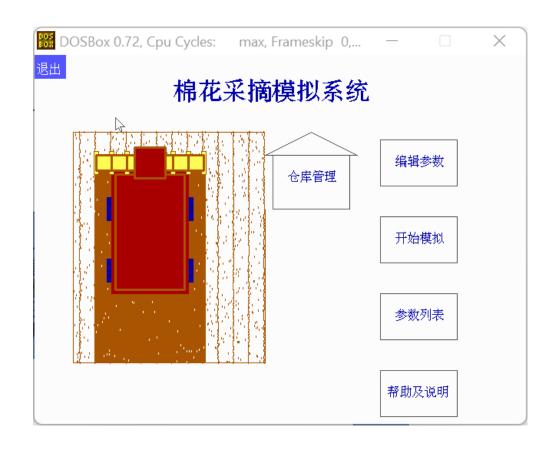
DOSBox 0.72, Cpu Cycles:	max, Frame	skip 0,	_	×
棉花枝	莫拟采集	長系统		
账号:				
密码:				
	登录			
忘记密码		注册	账号	
		B		

B DOSB	ox 0.72, Cpu Cycles:	max, Frameskip 0,	_	×
		注册账号		
	账号			
	密码			
A	确认密码			
	电话号码			
	返回	确	认	

DOSB	ox 0.72, Cpu Cycles:	max, Frameskip 0,	_		×
		忘记密码			
	账号				
₩.	电话号码				
	新密码				
	返回	确	认	ď	

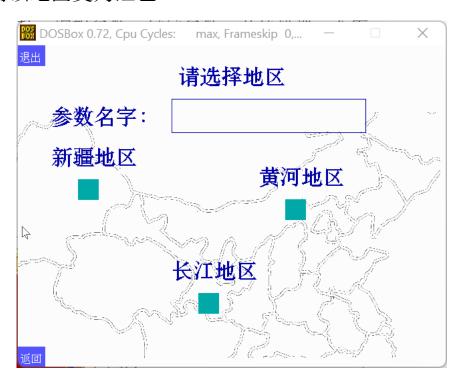
### 主界面:

运行之后的第一个界面,可以通过鼠标选择接下来的动作。 可选择的动作包括:编辑参数,过往参数,开始模拟,仓库 管理和帮助及说明。

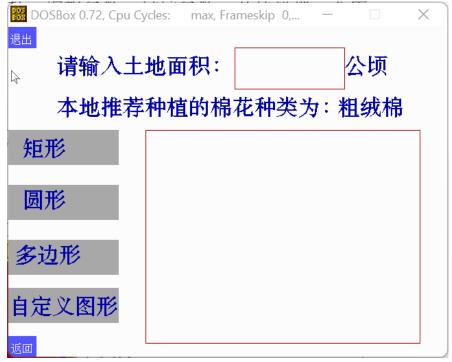


#### 编辑参数界面:

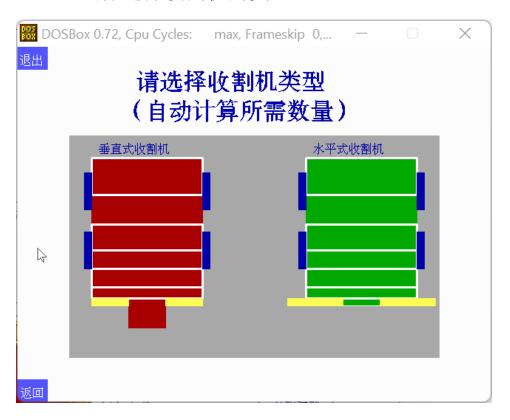
界面 1:选择产区,三个棉花产区,鼠标移动到地图上对应地区则该地区变为红色。



界面 2: 通过键盘输入土地面积,并显示推荐种植的棉花类型。

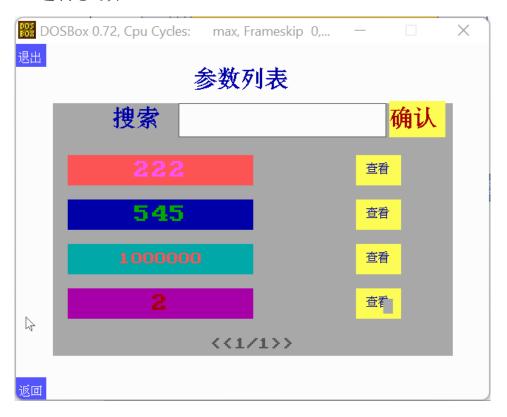


界面 3: 通过鼠标选择收割机的类型。



## 参数列表界面:

界面 1: 选择参数



界面 2: 选择参数后,选择返回可重新选择参数,查看可查 看该参数的内容。



界面 3: 查看参数后,可以修改指定的参数。

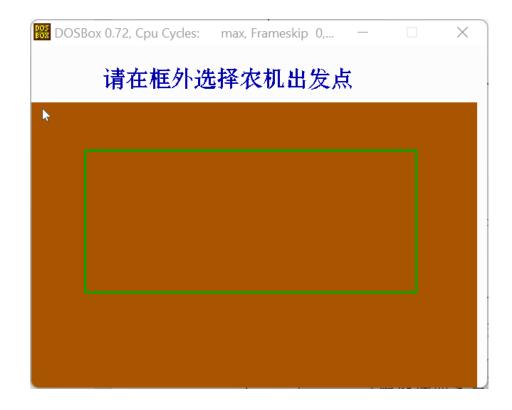


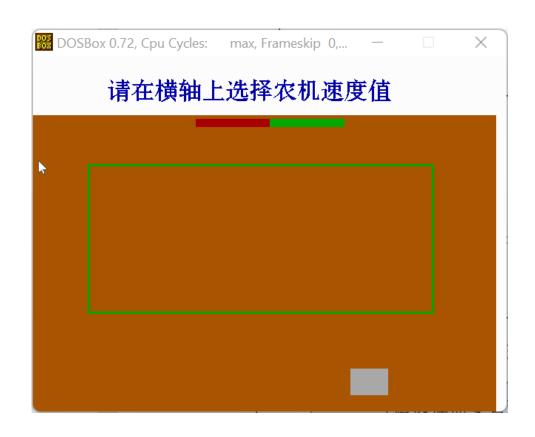
### 开始模拟界面:

界面 1: 选择需要模拟的参数。

DOSBox 0.72, Cpu Cycles:	max, Frameskip 0,	_	X
退出	清选择参数		
222		选择 ————————————————————————————————————	
545		选择	
1000000		选择	
2		选择	
	<<1/1>>		
返回			

界面 4: 选择出发点和农机设定速度





界面 3:播放采摘动画,并分别显示所需时间和收获量。





界面 4: 选择入库仓库并入库

DOSBox	c 0.72, Cpu Cycles:	max, Frameskip 0,	_	×				
<sub>退出</sub> 当前仓库列表								
	仓库名: ware() 1 棉花种类: 长绒棉	库存量: 4200	旽					
	仓库名: ware 02 棉花种类: 长绒棉	库存量 <b>: 5040</b>	庉					
	仓库名 <b>: here</b> 棉花种类 <b>: 细绒棉</b>	库存量 <b>: ()</b>	吨					
	仓库名:ware04棉花种类:长绒棉	库存量 <b>: 5040</b>	庉					
8	仓库名:ware05棉花种类:长绒棉	库存量 <b>: 5040</b>	庉					
返回								

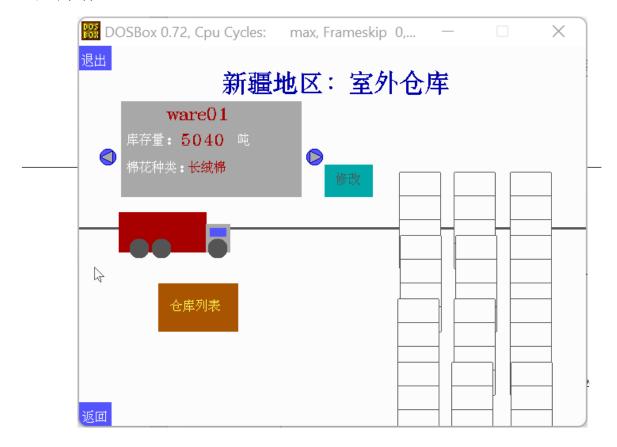
1

帮助与说明页:将鼠标置于各个按钮可以查看操作说明



仓储管理页:

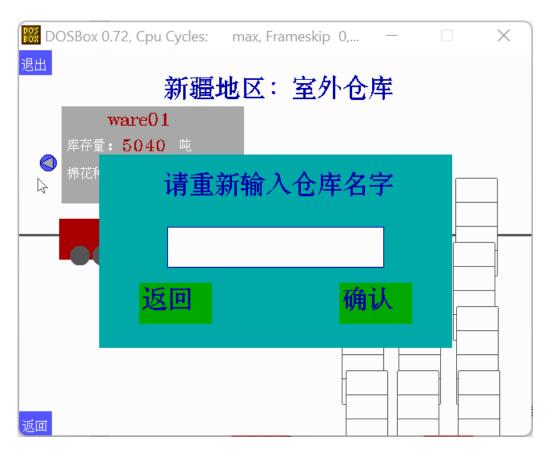
界面 1: 仓库主界面,显示当前参数设置地区及棉花种类对应库存量



界面 2: 仓库列表,可以更改当前仓库



界面 3: 仓库名字修改



界面4详细仓库信息及操作,采摘则进入模拟界面,出库则进入出库输入界面



界面 5: 出库输入界面



# 七、函数代码

#### #ifndef DSTART H <1>头文件部分 #define \_DSTART\_H\_ 1、 COMMON.H #include "COMMON.H" #ifndef\_COMMON\_H\_ #include "LOGFUN.H" #define \_COMMON\_H\_ #include "START.H" #include<stdio.h> #define x max 550 #include<stdlib.h> #define y\_max 320 #include<graphics.h> #define x start 50 #include<dos.h> #define y\_start 120 #include<malloc.h> #define point\_max 500 #include<bios.h> #define dense points max 500 #include<math.h> #include<string.h> #define tracktor\_num\_max 5 //#define delaytime 20 #include "mouse.h" #include<conio.h> #define tra time 80 #include ".\\HZK\\HZ.H" #define tracktor w 25 #define co pak w 10 //extern U ware here[5]; #define pick bar 600 extern int k; #define tracktor\_I 30 extern int mode; #define tra start 150 extern int mode1; #define tra\_start\_d 35 //extern char str[15]; #define pick ph 2.4 extern struct User\*h; #define Xinjinag\_har 2.1 void quit(void);//the leave button in every page #define Huanghe\_har 1.3 void skip(void); #define Chnagjiang\_har 1.1 #define p\_Up\_arrow 0x4800 void next(void); #define p Donw arrow 0x5000 void last(void); //void text\_input(char \*str,int x1,int y1,int x2,int #define p\_Enter 0x1C0D y2,int t\_x,int t\_y,int t\_size);//the input function for English characters and numbers void init\_tracktor01\_f(int x,int y);//initialize the //int ch\_to\_int(char \*str); tarcker 01 void input text(char \*id, int x, int y, int charnum, void init\_tracktor01\_b(int x,int y); int color, int flag); void init\_tracktor02\_f(int x,int y); void init\_tracktor02\_b(int x,int y); #endif void earth fill01(int x,int y);

2、

DSTART.H

void earth\_fill02(int x,int y);

void earth\_fill03(int x,int y);

```
void earth_cover01(int x,int y);
                                                                  #define _HOME_H_
    void earth cover02(int x,int y);
    void earth_cover03(int x,int y);
                                                                  #define ware full 29999
                                                                  #define num_ware 5
    void init tracktor01 r(int x,int y);
    void init tracktor01 l(int x,int y);
                                                                  void draw home01(void);//the page of n-w
    void init_tracktor02_r(int x,int y);
                                                             warehouse
    void init_tracktor02_l(int x,int y);
                                                                  //void draw_home00(void);//the page of Y-H
                                                             warehouse
    void init_picker_f(int x,int y);
                                                                  void warehouse_list(struct user_warehouse * w);
    void init picker b(int x,int y);
                                                                  void press warelist(int *re);
    void init_picker_r(int x,int y);
                                                                  void draw_warehouse(void);//draw the picture of
    void init_picker_l(int x,int y);
                                                             warehouse
                                                                  void cot_mount(int x,int y);//draw the cotton in
    void init_picker01_f(int x,int y);
                                                             warehouse
    void init_picker01_b(int x,int y);
                                                                  void draw_trunk(void);//draw the trunk int the
    void init_picker01_r(int x,int y);
                                                             page
    void init picker01 l(int x,int y);
                                                                          in warehouse(struct
                                                                  void
                                                                                                  user warehouse
                                                             *now);//draw the board which show the cotton in
    #endif
                                                             warehouse
                                                                  void detailed warehouse(long int count);//detail
     EDIT.H
                                                             message of warehouse
    #ifndef _EDIT_H_
                                                                  void press_detwarehouse(void);
    #define _EDIT_H_
                                                                        press_home(int *c_t);//add the press
                                                             moudule
    void edit01 screen(void);
                                                                  void out_warehouse(void);//page of the cotton
    int edit01(struct Parameter *abc);//The first in
                                                             out
edit:choose area
                                                                  void press_outware(void);
    void edit02_screen(struct Parameter *abc);
                                                                  void out_finish(void);
    int edit02(struct Parameter *abc);//The Second in
edit:define size and shape of the field
                                                                  typedef struct user_warehouse
    void edit03_screen(void);
                                                                  {
    int edit03(struct Parameter *abc);//Choose the
                                                                   char ware_name[15];
type of tractors and cotton
                                                                   int cotton type;
                                                                   long int total[3];
    void edit(void);
                                                                  }U_ware;
    #endif
                                                             #endif
                                                         7、
     START.H
                                                                   HELP.H
                                                                  #ifndef HELP H
                                                                  #define _HELP_H_
     PAST.H
                                                                  void draw help(void);
     HOME.H
                                                                  void
                                                                         help01(void);//Draw
                                                                                                the
                                                                                                       Help
                                                                                                               and
```

3

4、

5、

6、

#ifndef \_HOME\_H\_

```
Explanation page
                                                                char place;//产区
                                                             char shape;//土地形状
       void edit_help(void);
       void start_help(void);
                                                                char type;//收割机类型
       void past_help(void);
                                                             char S[10];//面积
       void ware help(void);
                                                             int x[dense points max];
                                                             int y[dense_points_max];//坐标
       void help(void);
                                                             int lenxy;//坐标数量
       #endif
                                                            }parameter;
8
        LOGFUN.H
                                                            typedef struct User
       #ifndef_LOGFUN_H_
                                                            {
       #define _LOGFUN_H_
                                                                char username[10];
                                                                char password[10];
       #include "HOME.H"
                                                                char phonenumber[12];
       #define PAR 10
                                                             struct Parameter parameter[PAR];//参数列表
       #define dense_points_max 500
                                                             int lenpar;//参数个数
                                                             struct user warehouse here[5];//仓库
       typedef struct Input
                                                            }user;
        int x1;//框的坐标
        int y1;
                                                            void
                                                                      wr user(char
                                                                                        username[],char
        int x2;
                                                        password[],char phonenumber[]);
        int y2;
                                                                 changepassword(char
                                                                                       username0[],char
        char string[19];
                                                        newpassword0[],char phonenumber0[]);
        char length;//输入的最大长度
                                                            int logg(char username0[],char password0[]);
        char cursor;//光标位置
        char flag;//0 代表不接受输入,1 代表可以接受
   输入
                                                        #endif
       }INPUT;
                                                    9
                                                             LOGIN.H
                                                            #ifndef LOGUSER H
                                                            #define _LOGUSER_H_
       // 产区,面积,形状(坐标),收割机类型,
   名字
                                                            void loginit_screen(void);
                                                            void loginit(void);
       // 产区 place: a--新疆; b--黄河; c--长江; 分
                                                            int logenter(void);
   别对应:长绒棉,细绒棉,粗绒棉
       // 面积 S: 123 (数字)
       // 形状 shape: a--矩形; b--圆形; c--多边形;
   d 自定义形状
                                                        #endif
       // 坐标 x 和坐标 y: c/d 才用
                                                     10
                                                             PARAMETER.H
       // 收割机类型 type: a 垂直/b 水平
       // 名字 name: abcdefg
                                                            #ifndef PARAMETE H
                                                            #define _PARAMETE_H_
       typedef struct Parameter
       {
                                                            //键盘输入函数
           char name[10];//名字
                                                            int input s(int x,int y,struct Input *word,int
```

```
#ifndef _REGISTER_H_
size, int mode);
                                                           #define _REGISTER_H_
    //判断名字是否为空 and 重复
    int judgename(char name[]);
                                                           void log_register(void);//注册
                                                           void register screen(void);
    //判断土地面积是否为数字和空
                                                           int
                                                                   username_same(char
                                                                                            *name,char
                                                       *number)://判断注册账号是否存在和电话号码是否
    int judgeS(char name[]);
                                                       正确
    //把参数写进 h, 然后把 h 写进文件里面
    void wr parameter(struct Parameter* abc);
                                                       #endif
    //修改产区
                                                   12、
                                                           RESET.H
    int changeplace(int par);
                                                           #ifndef RESET H
    //修改土地形状
                                                           #define _RESET_H_
    int changeshape(int par);
    //修改收割机类型
                                                           void reset_screen(void);
    int changetype(int par);
                                                           void reset(void);
    //修改面积
    int changeS(int par);
    //修改名字
    void changeparname(int par);
                                                           #endif
                                                   <2>源文件部分
    //删除参数
    void deletepar(int par);
                                                   1、
                                                          DSTART.C
    //把当前登录的用户重新写进文件
                                                           #include "DSTART.H"
    void wr_h(void);
                                                           // initialize the tarcker 01
    //把第二个参数赋给第一个参数
                                                           void init_tracktor01_f(int x, int y)
    void parcpy(struct Parameter *a,struct Parameter
                                                           {
*b);
                                                           int i;
                                                           setlinestyle(0, 0, 1);
    //选择参数
                                                           setcolor(DARKGRAY);
    int choosepar(void);
                                                           // The main rectangle
                                                           setfillstyle(1, RED);
    //搜索
                                                           bar(x + 3, y - 3, x + 22, y + 38);
    int search(char name[]);
                                                           rectangle(x + 3, y - 3, x + 22, y + 38);
                                                           setlinestyle(0, 0, 3);
                                                           rectangle(x + 6, y - 6, x + 19, y + 35);
    //改仓库名字
    void changewarename(int wi);
#endif
                                                           // The samll bars
                                                           setcolor(DARKGRAY);
    REGISTER.H
                                                           setlinestyle(0, 0, 1);
```

11、

```
setfillstyle(1, YELLOW);
                                                                        // The driver site
bar(x, y, x + 25, y - 3);
                                                                        setfillstyle(1, RED);
rectangle(x, y, x + 25, y + 3);
                                                                        bar(x + 7, y + 43, x + 18, y + 32);
for (i = x; i \le 25 + x; i += 2)
                                                                        rectangle(x + 7, y + 43, x + 18, y + 32);
{
                                                                        // The wheels
      bar(i - 1, y + 1, i + 1, y - 4);
      rectangle(i - 1, y + 1, i + 1, y - 4);
                                                                        setfillstyle(1, BLUE);
}
                                                                        bar(x + 1, y + 29, x + 3, y + 23);
                                                                        bar(x + 22, y + 29, x + 24, y + 23);
// The driver site
                                                                        bar(x + 1, y + 12, x + 3, y + 6);
setfillstyle(1, RED);
                                                                        bar(x + 22, y + 12, x + 24, y + 6);
bar(x + 7, y - 5, x + 18, y + 6);
                                                                        }
rectangle(x + 7, y - 5, x + 18, y + 6);
                                                                        void init tracktor01 r(int x, int y)
// The wheels
setfillstyle(1, BLUE);
                                                                        int i;
bar(x + 1, y + 9, x + 3, y + 15);
                                                                        setlinestyle(0, 0, 1);
                                                                        setcolor(DARKGRAY);
bar(x + 22, y + 9, x + 24, y + 15);
bar(x + 1, y + 26, x + 3, y + 32);
                                                                        // The main rectangle
bar(x + 22, y + 26, x + 24, y + 32);
                                                                        setfillstyle(1, RED);
}
                                                                        bar(x - 3, y + 3, x - 38, y + 22);
                                                                        rectangle(x - 3, y + 3, x - 38, y + 22);
// initialize the tracktor type 1
                                                                        setlinestyle(0, 0, 3);
void init_tracktor01_b(int x, int y)
                                                                        rectangle(x - 6, y + 6, x - 35, y + 19);
{
int i;
                                                                        // The samll bars
setlinestyle(0, 0, 1);
                                                                        setcolor(DARKGRAY);
setcolor(DARKGRAY);
                                                                        setlinestyle(0, 0, 1);
                                                                        setfillstyle(1, YELLOW);
// The main rectangle
setfillstyle(1, RED);
                                                                        bar(x, y, x - 3, y + 25);
bar(x + 3, y, x + 22, y + 41);
                                                                        rectangle(x, y, x - 3, y + 25);
rectangle(x + 3, y, x + 22, y + 41);
                                                                        for (i = y; i \le 25 + y; i += 2)
setlinestyle(0, 0, 3);
                                                                        {
rectangle(x + 6, y + 3, x + 19, y + 38);
                                                                              bar(x + 1, i - 1, x - 4, i + 1);
                                                                              rectangle(x + 1, i - 1, x - 4, i + 1);
// The samll bars
                                                                        }
setcolor(DARKGRAY);
                                                                        // The driver site
setlinestyle(0, 0, 1);
setfillstyle(1, YELLOW);
                                                                        setfillstyle(1, RED);
bar(x, y + 38, x + 25, y + 41);
                                                                        bar(x - 5, y + 7, x + 6, y + 18);
rectangle(x, y + 38, x + 25, y + 41);
                                                                        rectangle(x - 5, y + 7, x + 6, y + 18);
for (i = x; i \le 25 + x; i += 2)
{
                                                                        // The wheels
      bar(i - 1, y + 41, i + 1, y + 37);
                                                                        setfillstyle(1, BLUE);
      rectangle(i - 1, y + 42, i + 1, y + 37);
                                                                        bar(x - 9, y + 1, x - 15, y + 3);
}
                                                                        bar(x - 9, y + 22, x - 15, y + 24);
                                                                        bar(x - 26, y + 1, x - 32, y + 3);
```

```
bar(x - 26, y + 22, x - 32, y + 24);
                                                                        setfillstyle(1, GREEN);
}
                                                                        bar(x + 3, y - 3, x + 22, y + 38);
                                                                        rectangle(x + 3, y - 3, x + 22, y + 38);
void init_tracktor01_l(int x, int y)
                                                                        setlinestyle(0, 0, 3);
                                                                        rectangle(x + 6, y - 6, x + 19, y + 35);
int i;
setlinestyle(0, 0, 1);
                                                                        // The samll bars
setcolor(DARKGRAY);
                                                                        setcolor(DARKGRAY);
// The main rectangle
                                                                        setlinestyle(0, 0, 1);
setfillstyle(1, RED);
                                                                        setfillstyle(1, YELLOW);
bar(x, y + 3, x - 41, y + 22);
                                                                        bar(x, y, x + 25, y - 3);
rectangle(x, y + 3, x - 41, y + 22);
                                                                        rectangle(x, y, x + 25, y + 3);
setlinestyle(0, 0, 3);
                                                                        for (i = x; i \le 25 + x; i += 2)
rectangle(x - 3, y + 6, x - 38, y + 19);
                                                                             bar(i - 1, y + 1, i + 1, y - 4);
// The samll bars
                                                                             rectangle(i - 1, y + 1, i + 1, y - 4);
setcolor(DARKGRAY);
                                                                        }
setlinestyle(0, 0, 1);
setfillstyle(1, YELLOW);
                                                                        // The driver site
bar(x - 38, y, x - 41, y + 25);
                                                                        setfillstyle(1, GREEN);
rectangle(x - 38, y, x - 41, y + 25);
                                                                        bar(x + 7, y - 5, x + 18, y + 6);
for (i = y; i \le 25 + y; i += 2)
                                                                        rectangle(x + 7, y - 5, x + 18, y + 6);
{
      bar(x - 41, i - 1, x - 37, i + 1);
                                                                        // The wheels
      rectangle(x - 42, i - 1, x - 37, i + 1);
                                                                        setfillstyle(1, BLUE);
}
                                                                        bar(x + 1, y + 9, x + 3, y + 15);
                                                                        bar(x + 22, y + 9, x + 24, y + 15);
// The driver site
                                                                        bar(x + 1, y + 26, x + 3, y + 32);
setfillstyle(1, RED);
                                                                        bar(x + 22, y + 26, x + 24, y + 32);
bar(x - 43, y + 7, x - 32, y + 18);
                                                                       }
rectangle(x - 43, y + 7, x - 32, y + 18);
                                                                       // initialize the tracktor type 2 in backward
// The wheels
                                                                       void init_tracktor02_b(int x, int y)
setfillstyle(1, BLUE);
                                                                       {
bar(x - 29, y + 1, x - 23, y + 3);
                                                                        int i;
bar(x - 29, y + 22, x - 23, y + 24);
                                                                        setlinestyle(0, 0, 1);
bar(x - 12, y + 1, x - 6, y + 3);
                                                                        setcolor(DARKGRAY);
bar(x - 12, y + 22, x - 6, y + 24);
                                                                        // The main rectangle
}
                                                                        setfillstyle(1, GREEN);
                                                                        bar(x + 3, y, x + 22, y + 41);
// initialize the tarcker 02 in front
                                                                        rectangle(x + 3, y, x + 22, y + 41);
void init tracktor02 f(int x, int y)
                                                                        setlinestyle(0, 0, 3);
{
                                                                        rectangle(x + 6, y + 3, x + 19, y + 38);
int i;
setlinestyle(0, 0, 1);
                                                                        // The samll bars
                                                                        setcolor(DARKGRAY);
setcolor(DARKGRAY);
// The main rectangle
                                                                        setlinestyle(0, 0, 1);
```

```
setfillstyle(1, YELLOW);
                                                                        setfillstyle(1, GREEN);
bar(x, y + 38, x + 25, y + 41);
                                                                        bar(x - 5, y + 7, x + 6, y + 18);
rectangle(x, y + 38, x + 25, y + 41);
                                                                        rectangle(x - 5, y + 7, x + 6, y + 18);
for (i = x; i \le 25 + x; i += 2)
{
                                                                        // The wheels
      bar(i - 1, y + 41, i + 1, y + 37);
                                                                        setfillstyle(1, BLUE);
      rectangle(i - 1, y + 42, i + 1, y + 37);
                                                                        bar(x - 9, y + 1, x - 15, y + 3);
}
                                                                        bar(x - 9, y + 22, x - 15, y + 24);
                                                                        bar(x - 26, y + 1, x - 32, y + 3);
// The driver site
                                                                        bar(x - 26, y + 22, x - 32, y + 24);
setfillstyle(1, GREEN);
                                                                        }
bar(x + 7, y + 43, x + 18, y + 32);
rectangle(x + 7, y + 43, x + 18, y + 32);
                                                                        void init_tracktor02_l(int x, int y)
                                                                        {
// The wheels
                                                                        int i;
setfillstyle(1, BLUE);
                                                                        setlinestyle(0, 0, 1);
bar(x + 1, y + 29, x + 3, y + 23);
                                                                        setcolor(DARKGRAY);
bar(x + 22, y + 29, x + 24, y + 23);
                                                                        // The main rectangle
bar(x + 1, y + 12, x + 3, y + 6);
                                                                        setfillstyle(1, GREEN);
bar(x + 22, y + 12, x + 24, y + 6);
                                                                        bar(x, y + 3, x - 41, y + 22);
}
                                                                        rectangle(x, y + 3, x - 41, y + 22);
                                                                        setlinestyle(0, 0, 3);
void init_tracktor02_r(int x, int y)
                                                                        rectangle(x - 3, y + 6, x - 38, y + 19);
{
int i;
                                                                        // The samll bars
setlinestyle(0, 0, 1);
                                                                        setcolor(DARKGRAY);
setcolor(DARKGRAY);
                                                                        setlinestyle(0, 0, 1);
// The main rectangle
                                                                        setfillstyle(1, YELLOW);
setfillstyle(1, GREEN);
                                                                        bar(x - 38, y, x - 41, y + 25);
bar(x - 3, y + 3, x - 38, y + 22);
                                                                        rectangle(x - 38, y, x - 41, y + 25);
rectangle(x - 3, y + 3, x - 38, y + 22);
                                                                        for (i = y; i \le 25 + y; i += 2)
setlinestyle(0, 0, 3);
rectangle(x - 6, y + 6, x - 35, y + 19);
                                                                              bar(x - 41, i - 1, x - 37, i + 1);
                                                                              rectangle(x - 42, i - 1, x - 37, i + 1);
// The samll bars
                                                                        }
setcolor(DARKGRAY);
setlinestyle(0, 0, 1);
                                                                        // The driver site
setfillstyle(1, YELLOW);
                                                                        setfillstyle(1, GREEN);
bar(x, y, x - 3, y + 25);
                                                                        bar(x - 43, y + 7, x - 32, y + 18);
rectangle(x, y, x - 3, y + 25);
                                                                        rectangle(x - 43, y + 7, x - 32, y + 18);
for (i = y; i \le 25 + y; i += 2)
                                                                        // The wheels
      bar(x + 1, i - 1, x - 4, i + 1);
                                                                        setfillstyle(1, BLUE);
      rectangle(x + 1, i - 1, x - 4, i + 1);
                                                                        bar(x - 29, y + 1, x - 23, y + 3);
}
                                                                        bar(x - 29, y + 22, x - 23, y + 24);
                                                                        bar(x - 12, y + 1, x - 6, y + 3);
// The driver site
                                                                        bar(x - 12, y + 22, x - 6, y + 24);
```

```
}
                                                                        setfillstyle(1, BROWN);
                                                                        setcolor(WHITE);
    // earth filling after pick ,front
                                                                        bar(x - 1, y, x + 26, y + 41);
    void earth_fill01(int x, int y)
                                                                        // if(y\%4==0)
                                                                        //{
     setfillstyle(1, BROWN);
                                                                        // int x temp=rand()%25;
     setcolor(WHITE);
                                                                        //
                                                                             line(x+x_temp,y+41,x+x_temp,y+41);
     bar(x - 1, y, x + 25, y + 41);
                                                                        //}
     // if (y % 4 == 0)
                                                                       }
     //{
     // int x_temp = rand() % 25;
                                                                       // earth filling ,left and right
     // line(x + x_{temp}, y + 41, x + x_{temp}, y + 41);
                                                                       void earth cover02(int x, int y)
     //}
                                                                       {
     }
                                                                        setfillstyle(1, BROWN);
                                                                        setcolor(WHITE);
    // earth filling after pick,back
                                                                        bar(x - 1, y - 1, x - 41, y + 26);
    void earth_fill02(int x, int y)
                                                                        // if(y\%4==0)
                                                                        //{
     setfillstyle(1, BROWN);
                                                                        // int x temp=rand()%25;
     setcolor(WHITE);
                                                                             line(x+x_temp,y-1,x+x_temp,y-1);
     bar(x - 1, y - 1, x + 25, y + 41);
                                                                        //}
     // if (y % 4 == 0)
                                                                       }
     //{
     // int x_temp = rand() % 25;
                                                                       // earth filling , turn direction
          line(x + x_{temp}, y - 1, x + x_{temp}, y - 1);
                                                                       void earth_cover03(int x, int y)
     //
     //}
                                                                       {
    }
                                                                        int i;
                                                                        setfillstyle(1, BROWN);
    // earth filling after pick, turn direction
                                                                        setcolor(WHITE);
    void earth_fill03(int x, int y)
                                                                        bar(x - 1, y, x + 25, y + 50);
                                                                        // for(i=0;i<10;i++)
    {
     // int i;
                                                                        //{
     setfillstyle(1, BROWN);
                                                                        //
                                                                            int x_temp=rand()%25,y_temp=rand()%50;
     setcolor(WHITE);
                                                                        //
     bar(x - 1, y, x + 25, y + 50);
                                                                   line(x_temp+x,y_temp+y,x_temp+x,y_temp+y);
     // for (i = 0; i < 10; i++)
                                                                        //}
     //{
                                                                       }
     // int x_temp = rand() % 25, y_temp = rand() %
50;
                                                                       // initialize the picker 01
     // line(x_{temp} + x, y_{temp} + y, x_{temp} + x,
                                                                       void init_picker_f(int x,int y)
y_temp + y);
     //}
    }
                                                                        int i;
                                                                        setlinestyle(0, 0, 1);
    // earth filling ,front and back
                                                                        setcolor(DARKGRAY);
    void earth_cover01(int x, int y)
                                                                        // The main rectangle
    {
                                                                        setfillstyle(1, LIGHTGRAY);
```

```
bar(x + 3, y - 3, x + 22, y + 38);
                                                                        rectangle(x, y + 38, x + 25, y + 41);
rectangle(x + 3, y - 3, x + 22, y + 38);
                                                                        for (i = x; i \le 25 + x; i += 5)
setlinestyle(0, 0, 3);
                                                                        {
rectangle(x + 6, y - 6, x + 19, y + 35);
                                                                              bar(i - 1, y + 41, i + 1, y + 37);
                                                                              rectangle(i - 1, y + 42, i + 1, y + 37);
                                                                        }
// The samll bars
setcolor(DARKGRAY);
setlinestyle(0, 0, 1);
                                                                        // The driver site
setfillstyle(1, CYAN);
                                                                        setfillstyle(1, LIGHTBLUE);
bar(x, y, x + 25, y - 3);
                                                                        bar(x + 7, y + 43, x + 18, y + 32);
rectangle(x, y, x + 25, y + 3);
                                                                        rectangle(x + 7, y + 43, x + 18, y + 32);
for (i = x; i \le 25 + x; i += 5)
                                                                        // The wheels
{
      bar(i - 1, y + 1, i + 1, y - 4);
                                                                        setfillstyle(1, BLUE);
      rectangle(i - 1, y + 1, i + 1, y - 4);
                                                                        bar(x + 1, y + 29, x + 3, y + 23);
}
                                                                        bar(x + 22, y + 29, x + 24, y + 23);
                                                                        bar(x + 1, y + 12, x + 3, y + 6);
// The driver site
                                                                        bar(x + 22, y + 12, x + 24, y + 6);
setfillstyle(1, LIGHTBLUE);
bar(x + 7, y - 5, x + 18, y + 6);
rectangle(x + 7, y - 5, x + 18, y + 6);
                                                                       void init picker r(int x, int y)
// The wheels
                                                                        int i;
setfillstyle(1, BLUE);
                                                                        setlinestyle(0, 0, 1);
bar(x + 1, y + 9, x + 3, y + 15);
                                                                        setcolor(DARKGRAY);
bar(x + 22, y + 9, x + 24, y + 15);
                                                                        // The main rectangle
bar(x + 1, y + 26, x + 3, y + 32);
                                                                        setfillstyle(1, LIGHTGRAY);
bar(x + 22, y + 26, x + 24, y + 32);
                                                                        bar(x - 3, y + 3, x - 38, y + 22);
}
                                                                        rectangle(x - 3, y + 3, x - 38, y + 22);
                                                                        setlinestyle(0, 0, 3);
void init_picker_b(int x, int y)
                                                                        rectangle(x - 6, y + 6, x - 35, y + 19);
int i;
                                                                        // The samll bars
                                                                        setcolor(DARKGRAY);
setlinestyle(0, 0, 1);
setcolor(DARKGRAY);
                                                                        setlinestyle(0, 0, 1);
// The main rectangle
                                                                        setfillstyle(1, CYAN);
setfillstyle(1, LIGHTGRAY);
                                                                        bar(x, y, x - 3, y + 25);
bar(x + 3, y, x + 22, y + 41);
                                                                        rectangle(x, y, x - 3, y + 25);
rectangle(x + 3, y, x + 22, y + 41);
                                                                        for (i = y; i \le 25 + y; i += 5)
setlinestyle(0, 0, 3);
                                                                        {
rectangle(x + 6, y + 3, x + 19, y + 38);
                                                                              bar(x + 1, i - 1, x - 4, i + 1);
                                                                              rectangle(x + 1, i - 1, x - 4, i + 1);
// The samll bars
                                                                        }
setcolor(DARKGRAY);
setlinestyle(0, 0, 1);
                                                                        // The driver site
                                                                        setfillstyle(1, LIGHTBLUE);
setfillstyle(1, CYAN);
                                                                        bar(x - 5, y + 7, x + 6, y + 18);
bar(x, y + 38, x + 25, y + 41);
```

```
// initialize the picker after pick
rectangle(x - 5, y + 7, x + 6, y + 18);
                                                                       void init_picker01_f(int x,int y)
// The wheels
                                                                       {
setfillstyle(1, BLUE);
                                                                        int i;
bar(x - 9, y + 1, x - 15, y + 3);
                                                                        setlinestyle(0, 0, 1);
bar(x - 9, y + 22, x - 15, y + 24);
                                                                        setcolor(DARKGRAY);
bar(x - 26, y + 1, x - 32, y + 3);
                                                                        //The cotton package
bar(x - 26, y + 22, x - 32, y + 24);
                                                                        setfillstyle(1,WHITE);
                                                                        bar(x,y-1,x+tracktor_w,y-1+co_pak_w);
void init_picker_l(int x, int y)
                                                                        // The main rectangle
                                                                        setfillstyle(1, LIGHTGRAY);
int i;
                                                                        bar(x + 3, y - 3, x + 22, y + 38);
setlinestyle(0, 0, 1);
                                                                        rectangle(x + 3, y - 3, x + 22, y + 38);
setcolor(DARKGRAY);
                                                                        setlinestyle(0, 0, 3);
// The main rectangle
                                                                        rectangle(x + 6, y - 6, x + 19, y + 35);
setfillstyle(1, LIGHTGRAY);
                                                                        // The samll bars
bar(x, y + 3, x - 41, y + 22);
rectangle(x, y + 3, x - 41, y + 22);
                                                                        setcolor(DARKGRAY);
setlinestyle(0, 0, 3);
                                                                        setlinestyle(0, 0, 1);
rectangle(x - 3, y + 6, x - 38, y + 19);
                                                                        setfillstyle(1, CYAN);
                                                                        bar(x, y, x + 25, y - 3);
// The samll bars
                                                                        rectangle(x, y, x + 25, y + 3);
setcolor(DARKGRAY);
                                                                        for (i = x; i \le 25 + x; i += 5)
setlinestyle(0, 0, 1);
                                                                        {
setfillstyle(1, CYAN);
                                                                             bar(i - 1, y + 1, i + 1, y - 4);
bar(x - 38, y, x - 41, y + 25);
                                                                             rectangle(i - 1, y + 1, i + 1, y - 4);
rectangle(x - 38, y, x - 41, y + 25);
                                                                        }
for (i = y; i \le 25 + y; i += 5)
{
                                                                        // The driver site
      bar(x - 41, i - 1, x - 37, i + 1);
                                                                        setfillstyle(1, LIGHTBLUE);
      rectangle(x - 42, i - 1, x - 37, i + 1);
                                                                        bar(x + 7, y - 5, x + 18, y + 6);
}
                                                                        rectangle(x + 7, y - 5, x + 18, y + 6);
// The driver site
                                                                        // The wheels
setfillstyle(1, LIGHTBLUE);
                                                                        setfillstyle(1, BLUE);
bar(x - 43, y + 7, x - 32, y + 18);
                                                                        bar(x + 1, y + 9, x + 3, y + 15);
rectangle(x - 43, y + 7, x - 32, y + 18);
                                                                        bar(x + 22, y + 9, x + 24, y + 15);
                                                                        bar(x + 1, y + 26, x + 3, y + 32);
// The wheels
                                                                        bar(x + 22, y + 26, x + 24, y + 32);
setfillstyle(1, BLUE);
                                                                       }
bar(x - 29, y + 1, x - 23, y + 3);
bar(x - 29, y + 22, x - 23, y + 24);
                                                                       void init picker01 b(int x, int y)
bar(x - 12, y + 1, x - 6, y + 3);
                                                                       {
bar(x - 12, y + 22, x - 6, y + 24);
                                                                        int i;
                                                                        setlinestyle(0, 0, 1);
                                                                        setcolor(DARKGRAY);
```

```
//The cotton package
                                                                       bar(x - 3, y + 3, x - 38, y + 22);
setfillstyle(1,WHITE);
                                                                       rectangle(x - 3, y + 3, x - 38, y + 22);
bar(x,y+42,x+tracktor w,y+42-co pak w);
                                                                       setlinestyle(0, 0, 3);
                                                                       rectangle(x - 6, y + 6, x - 35, y + 19);
// The main rectangle
setfillstyle(1, LIGHTGRAY);
                                                                       // The samll bars
bar(x + 3, y, x + 22, y + 41);
                                                                       setcolor(DARKGRAY);
rectangle(x + 3, y, x + 22, y + 41);
                                                                       setlinestyle(0, 0, 1);
setlinestyle(0, 0, 3);
                                                                       setfillstyle(1, CYAN);
rectangle(x + 6, y + 3, x + 19, y + 38);
                                                                       bar(x, y, x - 3, y + 25);
                                                                       rectangle(x, y, x - 3, y + 25);
// The samll bars
                                                                       for (i = y; i \le 25 + y; i += 5)
setcolor(DARKGRAY);
                                                                       {
setlinestyle(0, 0, 1);
                                                                            bar(x + 1, i - 1, x - 4, i + 1);
setfillstyle(1, CYAN);
                                                                            rectangle(x + 1, i - 1, x - 4, i + 1);
bar(x, y + 38, x + 25, y + 41);
                                                                       }
rectangle(x, y + 38, x + 25, y + 41);
                                                                       // The driver site
for (i = x; i \le 25 + x; i += 5)
                                                                       setfillstyle(1, LIGHTBLUE);
      bar(i - 1, y + 41, i + 1, y + 37);
                                                                       bar(x - 5, y + 7, x + 6, y + 18);
      rectangle(i - 1, y + 42, i + 1, y + 37);
                                                                       rectangle(x - 5, y + 7, x + 6, y + 18);
}
                                                                       // The wheels
// The driver site
                                                                       setfillstyle(1, BLUE);
setfillstyle(1, LIGHTBLUE);
                                                                       bar(x - 9, y + 1, x - 15, y + 3);
                                                                       bar(x - 9, y + 22, x - 15, y + 24);
bar(x + 7, y + 43, x + 18, y + 32);
rectangle(x + 7, y + 43, x + 18, y + 32);
                                                                       bar(x - 26, y + 1, x - 32, y + 3);
                                                                       bar(x - 26, y + 22, x - 32, y + 24);
                                                                      }
// The wheels
setfillstyle(1, BLUE);
bar(x + 1, y + 29, x + 3, y + 23);
                                                                      void init picker01 l(int x, int y)
bar(x + 22, y + 29, x + 24, y + 23);
bar(x + 1, y + 12, x + 3, y + 6);
                                                                       int i;
bar(x + 22, y + 12, x + 24, y + 6);
                                                                       setlinestyle(0, 0, 1);
                                                                       setcolor(DARKGRAY);
}
                                                                       //The cotton package
void init picker01 r(int x, int y)
                                                                       setfillstyle(1,WHITE);
                                                                       bar(x-42,y,x-42+co_pak_w,y+tracktor_w);
int i;
setlinestyle(0, 0, 1);
                                                                       // The main rectangle
setcolor(DARKGRAY);
                                                                       setfillstyle(1, LIGHTGRAY);
//The cotton package
                                                                       bar(x, y + 3, x - 41, y + 22);
setfillstyle(1,WHITE);
                                                                       rectangle(x, y + 3, x - 41, y + 22);
bar(x+1,y,x+1-co_pak_w,y+tracktor_w);
                                                                       setlinestyle(0, 0, 3);
                                                                       rectangle(x - 3, y + 6, x - 38, y + 19);
// The main rectangle
                                                                       // The samll bars
setfillstyle(1, LIGHTGRAY);
```

```
setcolor(DARKGRAY);
                                                                      puthz(50, 90, "参数名字: ", 32, 32, BLUE);
 setlinestyle(0, 0, 1);
 setfillstyle(1, CYAN);
                                                                     // 按钮
 bar(x - 38, y, x - 41, y + 25);
                                                                      puthz(80 - 30, 150, "新疆地区", 32, 32,
 rectangle(x - 38, y, x - 41, y + 25);
                                                               BLUE);
                                                                      puthz(400 - 40, 150 + 30, "黄河地区", 32,
 for (i = y; i \le 25 + y; i += 5)
 {
                                                               32, BLUE);
      bar(x - 41, i - 1, x - 37, i + 1);
                                                                      puthz(250 - 20, 300 + 20, "长江地区", 32,
      rectangle(x - 42, i - 1, x - 37, i + 1);
                                                               32, BLUE);
 }
                                                                      setcolor(CYAN);
                                                                      bar(120 - 30, 200, 150 - 30, 230);
 // The driver site
                                                                      bar(440 - 40, 200 + 30, 470 - 40, 230 + 30);
 setfillstyle(1, LIGHTBLUE);
                                                                      bar(290 - 20, 350 + 20, 320 - 20, 380 + 20);
 bar(x - 43, y + 7, x - 32, y + 18);
                                                                      rectangle(230, 80, 520, 130);
 rectangle(x - 43, y + 7, x - 32, y + 18);
                                                                    }
 // The wheels
                                                                    int edit01(struct Parameter *abc)
 setfillstyle(1, BLUE);
 bar(x - 29, y + 1, x - 23, y + 3);
                                                                      INPUT name = {230, 80, 520, 130, "", 10, 0,
 bar(x - 29, y + 22, x - 23, y + 24);
                                                               0};
 bar(x - 12, y + 1, x - 6, y + 3);
                                                                      edit01 screen();
 bar(x - 12, y + 22, x - 6, y + 24);
                                                                      setfillstyle(1, MAGENTA); // 洋红色
                                                                      for (;;)
FDIT.C
                                                                      {
     #include "EDIT.H"
                                                                           newmouse(&MouseX,
                                                                                                        &MouseY,
     #include "COMMON.H"
                                                               &press);
     #include "IMAGE.h"
                                                                           input_s(233, 80, &name, 16, 0);
     #include "PARAMETE.H"
                                                                          if (mouse_press(100 - 30, 180, 170 - 30,
     #include "LOGFUN.H"
                                                               250) == 2) // 新疆
    #include "START.H"
                                                                          {
                                                                               setfillstyle(1, MAGENTA);
                                                                               bar(120 - 30, 200, 150 - 30, 230);
    void edit01 screen()
                                                                          }
      clrmous(MouseX, MouseY);
                                                                           else
      cleardevice();
                                                                          {
      setbkcolor(WHITE);
                                                                               setfillstyle(1, CYAN);
      bmp convert(".\\photo\\map.bmp",
                                                                               bar(120 - 30, 200, 150 - 30, 230);
".\\photo\\map.dbm");
      show_dbm(5, 100, ".\photo\map.dbm");
                                                                          if (mouse press(100 - 30, 180, 170 - 30,
      puthz(240, 30, "请选择地区", 32, 32,
                                                               250) == 1)
BLUE);
                                                                          {
                                                                               if (judgename(name.string) == 1)
      quit();
      last();
                                                                               {
      setcolor(12); // 淡红色
                                                                                     strcpy(abc->name,
                                                               name.string);
      // 参数名字
                                                                                     abc->place = 'a';
```

2、

```
return 0;
                                                                                  strcpy(abc->name,
               }
                                                              name.string);
          }
                                                                                  abc->place = 'c';
                                                                                  return 0;
          if (mouse_press(420 - 40, 180 + 30,
                                                                             }
490 - 40, 250 + 30) == 2) // 黄河
                                                                        }
          {
               setfillstyle(1, MAGENTA);
                                                                        // quit
               bar(440 - 40, 200 + 30, 470 - 40,
                                                                         if (mouse_press(0, 0, 40, 30) == 0 ||
230 + 30);
                                                              mouse press(0, 450, 40, 480) == 0)
          }
                                                                        {
          else
                                                                             MouseS = 0;
          {
                                                                        }
               setfillstyle(1, CYAN);
                                                                         if (mouse press(0, 0, 40, 30) == 2 | |
               bar(440 - 40, 200 + 30, 470 - 40,
                                                              mouse_press(0, 450, 40, 480) == 2)
230 + 30);
                                                                        {
                                                                             MouseS = 1;
          if (mouse_press(420 - 40, 180 + 30,
490 - 40, 250 + 30) == 1
                                                                         if (mouse press(0, 0, 40, 30) == 1)
          {
                                                                        {
               if (judgename(name.string) == 1)
                                                                             exit(0);
                                                                        }
                    strcpy(abc->name,
                                                                        // last
name.string);
                    abc->place = 'b';
                                                                         if (mouse_press(0, 450, 40, 480) == 1)
                    return 0;
               }
                                                                             return 1;
          }
                                                                         delay(15);
          if (mouse_press(270 - 20, 330 + 20,
                                                                    }
340 - 20, 400 + 20) == 2) // 长江
                                                                    // getchar();
          {
                                                                    // closegraph();
               setfillstyle(1, MAGENTA);
                                                                  }
               bar(290 - 20, 350 + 20, 320 - 20,
380 + 20);
                                                                  void edit02 screen(struct Parameter *abc)
          }
          else
                                                                    clrmous(MouseX, MouseY);
                                                                    cleardevice();
               setfillstyle(1, CYAN);
                                                                    setbkcolor(WHITE);
               bar(290 - 20, 350 + 20, 320 - 20,
380 + 20);
                                                                    quit();
                                                                    last();
          if (mouse_press(270 - 20, 330 + 20,
340 - 20, 400 + 20) == 1
                                                                    puthz(70, 40, "请输入土地面积: ", 32, 32,
                                                              BLUE);
          {
               if (judgename(name.string) == 1)
                                                                    puthz(490, 40, "公顷", 32, 32, BLUE);
               {
                                                                    puthz(70, 100, "本地推荐种植的棉花种类
```

```
为: ", 32, 32, BLUE);
                                                              &press);
                                                                        delay(15);
      if (abc->place == 'a')
                                                                        input_s(333, 30, &S, 16, 0);
     {
           puthz(480, 100, "粗绒棉", 32, 32,
                                                                        // 土地形状按钮,停留在上面
BLUE);
                                                                        if (mouse_press(0, 150, 160, 200) == 2)
                                                              // 矩形
     }
      else if (abc->place == 'b')
                                                                        {
                                                                             if (flagcan1 == 1)
          puthz(480, 100, "长绒棉", 32, 32,
                                                                             {
BLUE);
                                                                                  clrmous(MouseX, MouseY);
     }
                                                                                  MouseS = 1;
      else
                                                                                  setfillstyle(1, MAGENTA);
      {
                                                                                  bar(0, 150, 160, 200);
          puthz(480, 100, "细绒棉", 32, 32,
                                                                                 puthz(20, 160, "矩形", 32, 32,
BLUE);
                                                              BLUE);
                                                                                  setfillstyle(10, BROWN);
     }
                                                                                  rectangle(240,
                                                                                                  190,
                                                                                                        560,
     // 土地形状
                                                              420);
      setfillstyle(1, LIGHTGRAY);
                                                                                  bar(241, 191, 559, 419);
      bar(0, 150, 160, 200);
                                                                                  flagcan = 0;
      bar(0, 230, 160, 280);
      bar(0, 310, 160, 360);
                                                                                  flagcan1 = 0;
                                                                             }
      bar(0, 380, 160, 430);
      puthz(20, 160, "矩形", 32, 32, BLUE);
                                                                        }
                                                                         else if (mouse press(0, 230, 160, 280)
      puthz(20, 235, "圆形", 32, 32, BLUE);
      puthz(10, 315, "多边形", 32, 32, BLUE);
                                                              == 2) // 圆形
      puthz(0, 390, "自定义形状", 32, 32, BLUE);
                                                                        {
                                                                             if (flagcan1 == 1)
      setfillstyle(1, WHITE);
                                                                             {
      bar(200, 150, 600, 460);
                                                                                  clrmous(MouseX, MouseY);
      setcolor(RED);
                                                                                  MouseS = 1;
      rectangle(200, 150, 600, 460);
                                                                                  setfillstyle(1, MAGENTA);
      rectangle(330, 30, 490, 90);
                                                                                  bar(0, 230, 160, 280);
                                                                                 puthz(20, 235, "圆形", 32, 32,
    }
                                                              BLUE);
    int edit02(struct Parameter *abc)
                                                                                  setfillstyle(10, LIGHTGRAY);
                                                                                  circle(400, 305, 121);
     // int flag = 0; // 返回键判断
                                                                                  pieslice(400, 305, 0, 360,
     int flagcan = 0, flagcan1 = 1;
                                                              120);
      INPUT S = {330, 30, 490, 90, "", 6, 0, 0};
                                                                                  line(280, 305, 520, 305);
                                                                                  line(400, 185, 400, 425);
      edit02 screen(abc);
                                                                                  flagcan = 0;
      for (;;)
                                                                                  flagcan1 = 0;
      {
                                                                             }
                                                                        }
          newmouse(&MouseX,
                                       &MouseY,
```

```
{
           else if (mouse_press(0, 310, 160, 360)
== 2) // 多边形
                                                                                if (flagcan == 0)
           {
                                                                                {
                if (flagcan1 == 1)
                                                                                     clrmous(MouseX, MouseY);
                                                                                     MouseS = 0;
                     int dindian[8] = \{220, 270,
                                                                                     setfillstyle(1, LIGHTGRAY);
350, 270, 285, 170, 220, 270}, dindian2[10] = {560,
                                                                                     bar(0, 150, 160, 200);
420, 560, 330, 400, 330, 400, 400, 560, 420}; //
                                                                                    puthz(20, 160, "矩形", 32, 32,
200,150,600,460
                                                               BLUE);
                     clrmous(MouseX, MouseY);
                                                                                     bar(0, 380, 160, 430);
                                                                                     puthz(0, 390, "自定义图形",
                     MouseS = 1;
                     setfillstyle(1, MAGENTA);
                                                               32, 32, BLUE);
                     bar(0, 310, 160, 360);
                                                                                     bar(0, 310, 160, 360);
                     puthz(10, 315, "多边形", 32,
                                                                                     puthz(10, 315, "多边形", 32,
32, BLUE);
                                                               32, BLUE);
                    setfillstyle(10, LIGHTGRAY);
                                                                                     bar(0, 230, 160, 280);
                    fillpoly(4, dindian);
                                                                                    puthz(20, 235, "圆形", 32, 32,
                     fillpoly(5, dindian2);
                                                               BLUE);
                     setlinestyle(0, 0, 3);
                                                                                     setfillstyle(1, WHITE);
                    line(220, 440, 580, 170);
                                                                                     bar(200, 150, 600, 460);
                    flagcan = 0;
                                                                                     setcolor(RED);
                     flagcan1 = 0;
                                                                                     rectangle(200,
                                                                                                      150,
                                                                                                              600,
                }
                                                               460);
                                                                                     flagcan = 1;
           else if (mouse_press(0, 380, 160, 430)
                                                                                     flagcan1 = 1;
== 2) // 自定义图形
                                                                                }
                                                                           }
           {
                if (flagcan1 == 1)
                {
                                                                           if (mouse_press(0, 150, 160, 200) == 1)
                     clrmous(MouseX, MouseY);
                                                               // 矩形
                     MouseS = 1;
                                                                           {
                     setlinestyle(0, 0, 15);
                                                                                if (judgeS(S.string) == 1)
                     setfillstyle(1, MAGENTA);
                                                                                {
                     bar(0, 380, 160, 430);
                                                                                     strcpy(abc->S, S.string);
                     puthz(0, 390, "自定义图形",
                                                                                     abc->shape = 'a';
32, 32, BLUE);
                                                                                     return 0;
                     setfillstyle(1, LIGHTGRAY);
                                                                                }
                     arc(400, 230, -90, 180, 60);
                                                                           else if (mouse_press(0, 230, 160, 280)
                    line(400, 290, 400, 370);
                     setfillstyle(1, BLACK);
                                                               == 1) // 圆形
                     circle(400, 390, 10);
                                                                           {
                                                                                if (judgeS(S.string) == 1)
                    flagcan = 0;
                                                                                {
                     flagcan1 = 0;
                                                                                     strcpy(abc->S, S.string);
                                                                                     abc->shape = 'b';
                }
           }
                                                                                     return 0;
                                                                                }
           else
```

```
}
           else if (mouse_press(0, 300, 160, 350)
                                                                     void edit03_screen()
== 1) // 多边形
                                                                     {
                                                                      int dindian0[10] = {100, 150, 100, 200, 250,
           {
                                                                200, 250, 150, 100, 150};
                if (judgeS(S.string) == 1)
                                                                      clrmous(MouseX, MouseY);
                     strcpy(abc->S, S.string);
                                                                      cleardevice();
                     abc->shape = 'c';
                                                                      setbkcolor(WHITE);
                     select02(abc);
                     return 0;
                                                                      quit();
                }
                                                                      last();
           }
           else if (mouse_press(0, 370, 160, 420)
                                                                      puthz(160, 30, "请选择收割机类型", 32, 32,
== 1) // 自定义图形
                                                                BLUE);
           {
                                                                      puthz(140, 70, "(自动计算所需数量)", 32,
                if (judgeS(S.string) == 1)
                                                                32, BLUE);
                {
                     strcpy(abc->S, S.string);
                                                                      setfillstyle(1, LIGHTGRAY);
                     abc->shape = 'd';
                                                                      bar(70, 120, 570, 420);
                     select03(abc);
                     return 0;
                                                                      setfillstyle(1, RED);
               }
                                                                      bar(100, 150, 250, 350);
           }
                                                                      setcolor(0); // 白色
                                                                      setlinestyle(0, 0, 3);
           // quit
                                                                      drawpoly(5, dindian0);
           if (mouse press(0, 0, 40, 30) == 0 ||
                                                                      rectangle(100, 240, 250, 275);
mouse_press(0, 450, 40, 480) == 0)
                                                                      rectangle(100, 275, 250, 300);
                                                                      setlinestyle(0, 0, 5);
           {
                                                                      rectangle(100, 300, 250, 325);
                MouseS = 0;
                                                                      rectangle(100, 325, 250, 340);
           }
           if (mouse press(0, 0, 40, 30) == 2 ||
                                                                      setlinestyle(0, 0, 2);
mouse_press(0, 450, 40, 480) == 2)
                                                                      rectangle(150, 340, 200, 350);
           {
                                                                      bar(150, 350, 200, 380);
                MouseS = 1;
                                                                      setfillstyle(1, YELLOW);
                                                                      bar(100, 340, 150, 350);
           if (mouse press(0, 0, 40, 30) == 1)
                                                                      bar(200, 340, 250, 350);
           {
                                                                      setfillstyle(1, BLUE);
                exit(0);
                                                                      bar(90, 170, 100, 220);
           }
                                                                      bar(90 + 160, 170, 100 + 160, 220);
                                                                      bar(90, 250, 100, 300);
                                                                      bar(90 + 160, 250, 100 + 160, 300);
           if (mouse press(0, 450, 40, 480) == 1)
           {
                                                                      setfillstyle(1, GREEN);
                return 1;
                                                                      bar(100 + 290, 150, 250 + 290, 350);
           }
                                                                      setcolor(0); // 白色
                                                                      setlinestyle(0, 0, 3);
      }
    }
                                                                      rectangle(100 + 290, 150, 250 + 290, 200);
```

```
rectangle(100 + 290, 240, 250 + 290, 275);
                                                                                    flag = 1;
      rectangle(100 + 290, 275, 250 + 290, 300);
                                                                                }
      setlinestyle(0, 0, 5);
                                                                          }
      rectangle(100 + 290, 300, 250 + 290, 325);
                                                                           else if (mouse_press(70, 120, 300, 420)
      rectangle(100 + 290, 325, 250 + 290, 340);
                                                               == 1)
      setlinestyle(0, 0, 2);
                                                                          {
      rectangle(150 + 290, 340, 200 + 290, 350);
                                                                                abc->type = 'a';
      setfillstyle(1, YELLOW);
                                                                                return 0;
      bar(100 + 290 - 25, 340, 150 + 290, 350);
                                                                          }
      bar(200 + 290, 340, 250 + 290 + 25, 350);
                                                                           else if (mouse press(340, 120, 570,
                                                               420) == 2)
      setfillstyle(1, BLUE);
      bar(90 + 290, 170, 100 + 290, 220);
                                                                           {
      bar(90 + 160 + 290, 170, 100 + 160 + 290,
                                                                                if (flag == 0)
220);
                                                                                {
      bar(90 + 290, 250, 100 + 290, 300);
                                                                                     MouseS = 1;
      bar(90 + 160 + 290, 250, 100 + 160 + 290,
                                                                                    setcolor(RED);
300);
                                                                                     setlinestyle(0, 0, 5);
                                                                                     rectangle(340,
                                                                                                      120,
                                                                                                             570,
      puthz(110, 130, "垂直式收割机", 16, 16,
                                                               420);
BLUE);
      puthz(400, 130, "水平式收割机", 16, 16,
                                                                                    flag = 1;
BLUE);
                                                                                }
    }
                                                                           }
                                                                           else if (mouse_press(340, 120, 570,
    int edit03(struct Parameter *abc)
                                                               420) == 1)
                                                                          {
      int flag = 0;
                                                                                abc->type = 'b';
                                                                                return 0;
      edit03_screen();
                                                                          }
                                                                           else
      for (;;)
                                                                          {
      {
                                                                                if (flag == 1)
           newmouse(&MouseX,
                                        &MouseY,
                                                                                {
&press);
                                                                                     MouseS = 0;
           delay(15);
                                                                                    setlinestyle(0, 0, 5);
                                                                                     setcolor(LIGHTGRAY);
           // 收割机按钮 70,120,570,420
                                                                                     rectangle(340,
                                                                                                             570,
           if (mouse press(70, 120, 300, 420) ==
                                                               420);
2)
                                                                                     rectangle(70, 120, 300, 420);
           {
                                                                                    flag = 0;
                if (flag == 0)
                                                                                }
                                                                          }
                     MouseS = 1;
                     setcolor(RED);
                    setlinestyle(0, 0, 5);
                                                                           if (mouse\_press(0, 0, 40, 30) == 0)
                     rectangle(70, 120, 300, 420);
                                                                          {
                                                                                MouseS = 0;
```

```
edit02:
         if (mouse\_press(0, 0, 40, 30) == 2)
                                                             if (edit02(abc) == 1)
         {
              MouseS = 1;
                                                                 goto edit01;
         if (mouse\_press(0, 0, 40, 30) == 1)
                                                             }
         {
              exit(0);
                                                             if (edit03(abc) == 1)
         }
                                                                 goto edit02;
         // last
         if (mouse press(0, 450, 40, 480) == 0)
                                                             wr parameter(abc);
                                                             free(abc);
         {
              MouseS = 0;
                                                             // h->parameter[h->lenpar]=*abc;
                                                       }
         if (mouse_press(0, 450, 40, 480) == 2)
                                                3、
                                                       HELP.C
         {
                                                    #include "COMMON.H"
              MouseS = 1;
                                                    #include "HELP.H"
                                                    #include "LOGFUN.H"
         if (mouse\_press(0, 450, 40, 480) == 1)
         {
                                                    void draw help()
              return 1;
                                                    {
         }
     }
                                                        setfillstyle(1, 0);
                                                        bar(0, 0, 640, 480);
    }
    // 产区,面积,形状(坐标),收割机类
                                                        puthz(240, 30, "帮助及说明", 32, 32, BLUE);
型,名字
                                                        quit();
    // 产区 place: a--新疆; b--黄河; c--长江;
                                                        last();
分别对应: 粗绒棉, 长绒棉, 细绒棉
                                                        puthz(30, 90, "本程序作为棉花自动化采摘之
    // 面积 S: 123 (数字)
                                                模拟程序,可就我国三大棉花产区进行收割模拟,并计
    // 形状 shape: a--矩形; b--圆形; c--多边
                                                算其收成及模拟仓储过程。", 32, 32, BLUE);
形; d 自定义形状
    // 坐标 xyz: c/d 才用: x1,y1,x2,y2,x3,y3....
                                                        setfillstyle(1, MAGENTA);
    // 收割机类型 type: a 垂直/b 水平
                                                        bar(320 + 15 - 5, 300 - 10, 320 + 15 + 120 + 5,
    // 名字 name: abcdefg
                                                380);
                                                        puthz(320 + 15, 305, "编辑参数", 32, 32, BLUE);
    // 土地形状按钮,按下去
                                                        bar(320 - 15 + 5, 300 - 10, 320 - 15 - 120 - 5,
    void edit()
                                                380);
                                                        puthz(320 - 15 - 120, 305, "开始模拟", 32, 32,
     parameter
                   *abc
                                  (parameter
                                                BLUE);
*)malloc(sizeof(parameter));
                                                        bar(320 + 15 + 150 - 5, 300 - 10, 320 + 15 + 120 +
                                                150 + 5, 380);
    edit01:
                                                        puthz(320 + 15 + 150, 305, "参数列表", 32, 32,
     if (edit01(abc) == 1)
                                                BLUE);
                                                        bar(320 - 15 - 150 + 5, 300 - 10, 320 - 15 - 120 -
                                                150 - 5, 380);
         return;
     }
                                                        puthz(320 - 15 - 120 - 150, 305, "仓库管理", 32,
```

```
{
32, BLUE);
    }
                                                                                    ware_help();
    void help01()
                                                                                    flag = 0;
                                                                                    flag1 = 0;
         int i, flag = 0, flag1 = 1;
                                                                                }
                                                                           }
         cleardevice();
         setbkcolor(WHITE);
                                                                           else
         draw_help();
                                                                           {
                                                                                if (flag == 0)
         setlinestyle(0,0,5);
                                                                                {
         for (;;)
                                                                                    clrmous(MouseX, MouseY);
                                                                                    setfillstyle(1, 0);
         {
              newmouse(&MouseX, &MouseY, &press);
                                                                                    bar(15, 85, 640, 240);
              if (mouse press(320 + 15 - 5, 300 - 10, 320
                                                                                    puthz(25, 90, "本程序作为棉花
                                                             自动化采摘之模拟程序, 可就我国三大棉花产区进行收
+ 15 + 120 + 5, 380) == 2) // 编辑参数
                                                             割模拟,并计算其收成及模拟仓储过程。", 32, 32, BLUE);
              {
                  if (flag1 == 1)
                                                                                    setfillstyle(1, MAGENTA);
                                                                                    bar(320 + 15 - 5, 300 - 10, 320 +
                  {
                       edit help();
                                                             15 + 120 + 5, 380;
                       flag = 0;
                                                                                    puthz(320 + 15, 305, "编辑参数",
                       flag1 = 0;
                                                             32, 32, BLUE);
                  }
                                                                                    bar(320 - 15 + 5, 300 - 10, 320 -
              }
                                                             15 - 120 - 5, 380);
              else if (mouse press(320 - 15 - 120 - 5, 300
                                                                                    puthz(320 - 15 - 120, 305, "开始
- 10, 320 - 15 + 5, 380) == 2) // 开始模拟
                                                             模拟", 32, 32, BLUE);
                                                                                    bar(320 + 15 + 150 - 5, 300 - 10,
                                                             320 + 15 + 120 + 150 + 5, 380);
                  if (flag1 == 1)
                                                                                    puthz(320 + 15 + 150, 305, "参数
                       start_help();
                                                             列表", 32, 32, BLUE);
                       flag = 0;
                                                                                    bar(320 - 15 - 150 + 5, 300 - 10,
                       flag1 = 0;
                                                             320 - 15 - 120 - 150 - 5, 380);
                  }
                                                                                    puthz(320 - 15 - 120 - 150, 305, "
                                                             仓库管理", 32, 32, BLUE);
              }
              else if (mouse_press(320 + 15 + 150 - 5, 300
                                                                                    flag = 1;
- 10, 320 + 15 + 120 + 150 + 5, 380) == 2) // 参数列表
                                                                                    flag1 = 1;
                                                                                }
                  if (flag1 == 1)
                                                                           }
                  {
                       past_help();
                                                                           if (mouse_press(0, 0, 40, 30) == 2 ||
                       flag = 0;
                                                             mouse_press(0, 450, 40, 480) == 2)
                       flag1 = 0;
                                                                           {
                  }
                                                                                MouseS = 1;
              }
             else if (mouse_press(320 - 15 - 120 - 150 - 5,
                                                                           if (mouse\_press(0, 0, 40, 30) == 1)
300 - 10, 320 - 15 - 150 + 5, 380) == 2) // 仓库管理
              {
                                                                                exit(0);
                  if (flag1 == 1)
                                                                           }
```

```
// last
                                                            {
                                                                clrmous(MouseX, MouseY);
             else if (mouse_press(0, 450, 40, 480) == 1)
                                                                setfillstyle(1, 0);
                 return;
                                                                bar(15, 85, 640, 240);
                                                                puthz(25, 90, "点击参数列表按钮可以查看已
             if (mouse_press(0, 0, 40, 30) == 0 ||
                                                       经创建了的参数,并对他们进行修改和删除等操作。",32,
mouse press(0, 450, 40, 480) == 0)
                                                       32, BLUE);
                                                                setcolor(CYAN);
            {
                 MouseS = 0;
                                                                rectangle(15, 85, 635, 240);
            }
                                                                setfillstyle(1, CYAN);
            // quit
                                                                bar(320 + 15 + 150 - 5, 300 - 10, 320 + 15 + 120 +
             delay(15);
                                                       150 + 5, 380);
        }
                                                                puthz(320 + 15 + 150, 305, "参数列表", 32, 32,
    }
                                                       BLUE);
    void edit_help(void)
                                                            void ware_help(void)
    {
                                                            {
        clrmous(MouseX, MouseY);
                                                                clrmous(MouseX, MouseY);
        setfillstyle(1, 0);
                                                                setfillstyle(1, 0);
        bar(15, 85, 640, 240);
                                                                bar(15, 85, 640, 240);
        puthz(25, 90, "选择编辑参数按钮即可创建新
                                                                puthz(25, 90, "点击仓库管理按钮即可对仓库
的参数。参数的数据包括:棉花产区、收割机类型、土
                                                       进行管理,可以进行出库和入库等操作",32,32,BLUE);
地的面积和形状等。", 32, 32, BLUE);
                                                                setcolor(LIGHTGRAY);
        setcolor(RED);
                                                                rectangle(15, 85, 635, 240);
        rectangle(15, 85, 635, 240);
                                                                setfillstyle(1, LIGHTGRAY);
                                                                bar(320 - 15 - 150 + 5, 300 - 10, 320 - 15 - 120 -
        setfillstyle(1, RED);
        bar(320 + 15 - 5, 300 - 10, 320 + 15 + 120 + 5,
                                                       150 - 5, 380);
380);
                                                                puthz(320 - 15 - 120 - 150, 305, "仓库管理", 32,
        puthz(320 + 15, 305, "编辑参数", 32, 32, BLUE);
                                                       32, BLUE);
                                                            }
    }
    void start help(void)
                                                            void help(void)
        clrmous(MouseX, MouseY);
                                                            {
        setfillstyle(1, 0);
                                                                help01();
        bar(15, 85, 640, 240);
                                                            }
        puthz(25, 90, "点击开始模拟按钮即可选择已
经创建的参数进行模拟采摘,将会播放采摘动画并显示
                                                       4、
                                                               HOME.C
采摘所需时间。", 32, 32, BLUE);
        setcolor(LIGHTRED);
                                                                   #include "COMMON.H"
        rectangle(15, 85, 635, 240);
                                                                   #include "START.H"
        setfillstyle(1, LIGHTRED);
                                                                   #include "LOGFUN.H"
        bar(320 - 15 + 5, 300 - 10, 320 - 15 - 120 - 5,
                                                                   #include "PARAMETE.H"
                                                                   #include "HOME.H"
380);
        puthz(320 - 15 - 120, 305, "开始模拟", 32, 32,
                                                                   #include "PAST.H"
BLUE);
                                                                   char str[15];
                                                                   int k;
```

void past help(void)

```
//
                                         U_ware
                                                                   if(c_tal>ware_full||c_tal<0)
here[5]={"ware01",0,{100,200,300},"ware02",1,{1
000,2000,3000},\
                                                                        c_tal=ware_full;
                                                                   }
    //
"ware03",2,{123,456,789},"ware04",1,{1234,4545
                                                                       //int num;
,234},"ware05",0,{34535,3423,6465}};
                                                                   settextstyle(3,0,4);
                                                                       cleardevice();
    //the page of n-w warehouse
                                                                       setbkcolor(WHITE);
    void draw_home01()
                                                                    draw_warehouse();
                                                                    draw trunk();
            i,type,location=0;//1
     int
                                   means
                                             the
                                                                   last();
norwestern,0 means others
                                                                   if(location==1)
     long int temp, c_tal;
                                                                   {
                                                                        puthz(180,30,"新疆地区: 室外仓库
      clrmous(MouseX,MouseY);
      for(i=0;i<PAR;i++)
                                                             ",32,32,BLUE);
      {
                                                                   }
                                                                   else
 if(strcmp(h->parameter[i].name,"\0")==0)
                                                                        puthz(120,30,"黄河、长江流域:室内
          {
                                                             仓库",32,32,BLUE);
               break;
          }
                                                                        setfillstyle(1,DARKGRAY);
     }
                                                                        bar(320,90,800,120);
                                                                        setfillstyle(1,LIGHTBLUE);
      settextjustify(0,2);
                                                                        bar(350,120,360,500);
     switch (h->parameter[i].place)
                                                                   if(strcmp(str,"\0"))
      {
          case 'a':
                                                                   {
          {
                                                                        temp=atoi(str);
               location=1;
                                                                        if(temp>c_tal)
               break;
                                                                        {
          }
                                                                             temp=c_tal;
          case 'b':
          {
                                                                        h->here[k].total[type]-=temp;
               location=0;
                                                                        for(i=0;i<15;i++)
               break;
          }
                                                                             str[i]='\0';
          case 'c':
                                                                        }
          {
               location=0;
                                                                    in_warehouse(h->here+k);
               break;
                                                                    quit();
          }
                                                                    clrmous(MouseX,MouseY);
                                                                   //wr h();
          default:
                                                                   // for(;;)
               break;
                                                                   //{
     }
                                                                   //
      type=h->here[k].cotton_type,temp=0;
                                                               newmouse(&MouseX,&MouseY,&press);
      c_tal=h->here[k].total[type];
                                                                        press_home(&(here[k].cotton_type));
```

```
//
          delay(15);
                                                                      if(*c_t<0)
     //}
    }
                                                                           *c_t=2;
    //add the press moudule
                                                                      // draw home01();
    void press_home(int *c_t)
                                                                      mode1=-1;
                                                                 }
                                                                 if(mouse_press(287,130,307,150)==1)
     if(mouse_press(0,0,40,30)==0||mouse_pre
ss(53,90,280,190)==0||mouse_press(26,130,46,1
50)==0\
                                                                      (*c_t)++;
      ||mouse_press(287,130,307,150)==0||mo
                                                                      if(*c_t>2)
use_press(100,300,200,360)==0||mouse_press(0,
450,40,480)==0\
                                                                           *c_t=0;
      ||mouse press(310,150,370,190)==0)
     {
                                                                      // draw_home01();
          MouseS=0;
                                                                      mode1=-1;
     }
      if(mouse_press(0,0,40,30)==2||mouse_pre
                                                                 if(mouse_press(310,150,370,190)==1)
ss(53,90,280,190)==2||mouse_press(26,130,46,1
50)==2\
                                                                      changewarename(k+1);
      ||mouse press(287,130,307,150)==2||mo
                                                                      mode1=-1;
use_press(100,300,200,360)==2||mouse_press(0,
450,40,480)==2\
                                                                 if(mouse_press(53,90,280,190)==1)
      ||mouse_press(310,150,370,190)==2)
     {
                                                                      //
                                                           detailed_warehouse(here[k].total[*c_t]);
          MouseS=1;
     }
                                                                      mode1=2;
     if(mouse_press(0,450,40,480)==1)
                                                                 }
                                                                }
          mode=0;
          mode1=0;
                                                                /*void draw_home00()
     }
     if(mouse\_press(0,0,40,30)==1)
                                                                     int i;
                                                                    //int num;
                                                                     cleardevice();
          // draw_wel();
          wr_h();
                                                                     setbkcolor(WHITE);
          free(h);
          exit(0);
                                                                 quit();
     }
                                                                     mouseinit();
     if(mouse_press(100,300,200,360)==1)
                                                                 for(i=0;i<1000;i++)
          // warehouse list(here,5);
          mode1=1;
                                                             newmouse(&MouseX,&MouseY,&press);
                                                                      delay(4);
     if(mouse_press(26,130,46,150)==1)
                                                                 }
     {
                                                                }*/
          (*c_t)--;
```

```
fillellipse(41-5,140,10,10);
    //draw the board which show the cotton in
warehouse
                                                                  fillellipse(292+5,140,10,10);
    void in_warehouse(U_ware* now)
                                                                  setfillstyle(1,LIGHTGRAY);
                                                                  fillpoly(3,arr1);
      char str1[8];
                                                                  fillpoly(3,arr2);
      int
arr1[6]={32-5,140,47-5,132,47-5,148},arr2[6]={30
                                                                 }
1+5,140,286+5,132,286+5,148};
      int type,count;
                                                                 void warehouse_list(U_ware *w)
     type=now->cotton type,count=now->total[
                                                                 {
                                                                  int i;
type];
      clrmous(MouseX,MouseY);
                                                                  cleardevice();
      setfillstyle(1,LIGHTGRAY);
                                                                  setbkcolor(WHITE);
      setlinestyle(0,0,1);
                                                                  clrmous(MouseX,MouseY);
      bar(53,70,280,190);
                                                                  setfillstyle(1,LIGHTGRAY);
      setfillstyle(1,CYAN);
                                                                  settextjustify(0,2);
      bar(310,150,370,190);
                                                                  puthz(220,30,"当前仓库列表",32,32,BLUE);
                                                                  bar(100,100,540,400);
      puthz(324,160,"修改",16,16,DARKGRAY);
      setcolor(RED);
      settextstyle(1,0,3);
                                                                  settextstyle(1,0,3);
      outtextxy(110,70,now->ware name);
                                                                  setfillstyle(1,WHITE);
      puthz(60,110,"库存量: ",16,16,WHITE);
                                                                  for(i=0;i<num_ware;i++)</pre>
      puthz(200,110,"吨",16,16,WHITE);
                                                                  {
      puthz(60,145,"棉花种类: ",16,16,WHITE);
                                                                       char str[15];
      setfillstyle(1,BROWN);
      bar(100,300,200,360);
                                                            up=100+60*i,down=160+i*60,type=w[i].cotton ty
      puthz(115,320,"仓库列表",16,16,YELLOW);
                                                            pe;
      switch (type)
                                                                       bar(100+2,up+2,540-2,down-2);
      {
                                                                       setlinestyle(0,0,1);
      case 0:
                                                                       setcolor(LIGHTBLUE);
          puthz(138,145,"长绒棉",16,16,RED);
                                                                       rectangle(100+3,up+3,540-3,down-3);
          break;
                                                                       setcolor(RED);
      case 1:
                                                                       puthz(104,up+10," 仓 库 名 :
          puthz(138,145,"细绒棉",16,16,RED);
                                                            ",16,16,DARKGRAY);
          break;
                                                                       outtextxy(168,up+5,w[i].ware name);
      case 2:
                                                                       puthz(104,up+30," 棉 花 种 类:
          puthz(138,145,"粗绒棉",16,16,RED);
                                                            ",16,16,DARKGRAY);
                                                                       puthz(320,up+30," 库 存 量:
          break;
      default:
                                                            ",16,16,DARKGRAY);
          break;
                                                                       puthz(510,up+30,"
                                                                                                          吨
                                                             ",16,16,DARKGRAY);
      }
      setcolor(RED);
                                                                       switch (type)
      itoa(count,str1,10);
                                                                       {
      settextstyle(1,0,2);
                                                                            case 0:
      outtextxy(130,105,str1);
                                                                            {
      setfillstyle(1,LIGHTBLUE);
      setcolor(BLUE);
                                                              if(w[i].total[type]<=ware full)
```

```
press_warelist(num_ware);
 itoa(w[i].total[type],str,10);
                                                                    //
                                                                         delay(15);
                    else
                                                                    //}
                         itoa(ware_full,str,10);
                                                                   }
                    puthz(184,up+30," 长 绒 棉
",16,16,RED);
                                                                   void press_warelist(int *re)
                    outtextxy(384,up+25,str);
                                                                   {
                    break;
                                                                    int i;
               }
                                                                    for(i=0;i<num_ware;i++)</pre>
               case 1:
                                                                         int up=100+60*i,down=160+i*60;
 if(w[i].total[type]<=ware_full)</pre>
                                                                if(mouse_press(100+2,up+2,540-2,down-2)==0)
                                                                         {
 itoa(w[i].total[type],str,10);
                                                                              MouseS=0;
                    else
                                                                              continue;
                         itoa(ware_full,str,10);
                                                                         }
                    puthz(184,up+30," 细 绒 棉
                                                                         else
",16,16,RED);
                                                              if(mouse_press(100+2,up+2,540-2,down-2)==2)
                    outtextxy(384,up+25,str);
                                                                         {
                    break;
                                                                              MouseS=1;
               }
                                                                              return;
               case 2:
                                                                         }
                                                                         else
                                                              if(mouse_press(100+2,up+2,540-2,down-2)==1)
 if(w[i].total[type]<=ware full)</pre>
                                                                         {
                                                                              k=i;
 itoa(w[i].total[type],str,10);
                                                                              // draw_home01();
                                                                              *re=1;
                    else
                         itoa(ware_full,str,10);
                                                                              mode1=0;
                    puthz(184,up+30," 粗 绒 棉
                                                                              return;
",16,16,RED);
                                                                         }
                    outtextxy(384,up+25,str);
                    break;
                                                                    if(mouse_press(0,0,40,30)==0||mouse_pre
               }
                                                              ss(0,450,40,480)==0)
                                                                    {
               default:
                                                                         MouseS=0;
                    break;
          }
                                                                     if(mouse_press(0,0,40,30)==2||mouse_pre
      }
                                                              ss(0,450,40,480)==2)
      quit();
                                                                    {
      last();
                                                                         MouseS=1;
      // while(1)
                                                                    if(mouse\_press(0,450,40,480)==1)
      //{
      //
                                                                         // draw_home01();
 newmouse(&MouseX,&MouseY,&press);
                                                                         mode1=0;
```

```
setfillstyle(1,LIGHTBLUE);
 else if(mouse_press(0,0,40,30)==1)
                                                             bar(140,320,240,380);
                                                             setfillstyle(1,RED);
                                                             bar(380,320,480,380);
      // draw_wel();
      wr h();
                                                             puthz(155,333,"采摘",32,32,WHITE);
                                                             puthz(395,333,"出库",32,32,WHITE);
      free(h);
      exit(0);
 }
                                                             quit();
}
                                                             // for(;;)
                                                             //{
                                                             //
//detail message of warehouse
                                                         newmouse(&MouseX,&MouseY,&press);
void detailed_warehouse(long int count)
                                                                  press_detwarehouse(count);
                                                                  delay(15);
 char str1[8];
                                                             //}
 cleardevice();
                                                            }
 setbkcolor(WHITE);
 clrmous(MouseX,MouseY);
 setfillstyle(1,LIGHTGRAY);
                                                            void press detwarehouse()
 puthz(220,30,"当前仓储信息",32,32,BLUE);
                                                            {
 bar(100,100,540,300);
                                                             if(mouse press(0,0,40,30)==0||mouse pre
 puthz(120,130,"库存量: ",32,32,WHITE);
                                                       ss(140,320,240,380)==0||mouse_press(380,320,4
 puthz(400,130,"吨",32,32,WHITE);
                                                       80,380)==0\
 puthz(120,220,"棉花种类: ",32,32,WHITE);
                                                             ||mouse_press(0,450,40,480)==0)
 switch (h->here[k].cotton_type)
                                                             {
 {
                                                                  MouseS=0;
 case 0:
      puthz(280,220,"长绒棉",32,32,RED);
                                                             if(mouse_press(0,0,40,30)==2||mouse_pre
                                                       ss(140,320,240,380)==2||mouse_press(380,320,4
      break;
                                                       80,380)==2\
                                                             ||mouse_press(0,450,40,480)==2)
 case 1:
      puthz(280,220,"细绒棉",32,32,RED);
                                                             {
      break;
                                                                  MouseS=1;
                                                             if(mouse press(0,0,40,30)==1)
 case 2:
      puthz(280,220,"粗绒棉",32,32,RED);
      break;
                                                                  // draw_wel();
                                                                  wr h();
 default:
                                                                  free(h);
                                                                  exit(0);
      break;
 itoa(count,str1,10);
                                                             if(mouse press(0,450,40,480)==1)
 setcolor(RED);
 settextstyle(1,0,4);
                                                                  // draw_home01();
 outtextxy(250,125,str1);
                                                                  mode1=0;
 last();
                                                             if(mouse press(140,320,240,380)==1)
```

```
{
                                                           ss(130,150,510,250)==0||mouse_press(270,320,3
                                                           70,380)==0\
         // draw_simu01(x_max,y_max,5);
         mode=3;
                                                                 ||mouse_press(0,450,40,480)==0)
         mode1=0;
                                                                 {
    }
                                                                      MouseS=0;
    if(mouse_press(380,320,480,380)==1)
    {
                                                                 if(mouse press(0,0,40,30)==2||mouse pre
         // out_warehouse(count);
                                                           ss(130,150,510,250)==2||mouse_press(270,320,3
         mode1=3;
                                                           70,380)==2\
    }
                                                                 ||mouse press(0,450,40,480)==2||
   }
                                                                 {
                                                                      MouseS=1;
   //page of the cotton out
   void out_warehouse()
                                                                 if(mouse\_press(0,0,40,30)==1)
    //int out;
                                                                     // draw_wel();
    clrmous(MouseX,MouseY);
                                                                     wr_h();
    //int kick=0;
                                                                     free(h);
    cleardevice();
                                                                      exit(0);
    setbkcolor(WHITE);
    setfillstyle(1,LIGHTGRAY);
                                                                 if(mouse press(0,450,40,480)==1)
    puthz(220,30,"请输入出库量",32,32,BLUE);
                                                                      // detailed_warehouse(count);
    setfillstyle(1,LIGHTGRAY);
                                                                      mode1=2;
    bar(100,100,540,300);
                                                                 }
                                                                 // if(mouse press(0,0,40,30)==1)
    setfillstyle(1,WHITE);
    bar(130,150,510,250);
                                                                 //{
    puthz(460,190,"吨",32,32,BLUE);
                                                                 //
                                                                     // detailed_warehouse(count);
    setfillstyle(1,RED);
                                                                 //
                                                                      mode=2;
    bar(270,320,370,380);
                                                                 //}
    puthz(285,333,"完成",32,32,WHITE);
                                                                 if(mouse press(130,150,510,250)==1)
    last();
    quit();
                                                            input_text(str,140,190,15,DARKGRAY,1);
    // for(;;)
                                                                      return;
    //{
                                                                 }
    //
                                                                 if(mouse_press(270,320,370,380)==1)
newmouse(&MouseX,&MouseY,&press);
         press_outware(count,str);
                                                                      out_finish();
    //
         delay(15);
    //}
                                                                 //return out;
    //return out;
   }
                                                               void out_finish()
   void press_outware()
                                                                 clrmous(MouseX,MouseY);
    if(mouse_press(0,0,40,30)==0||mouse_pre
                                                                 setfillstyle(1,WHITE);
```

```
bar(200,160,430,240);
                                                               void cot_mount(int x,int y)
 setcolor(BLUE);
 setlinestyle(0,0,3);
                                                                int i,d_y=y;
 rectangle(200,160,430,240);
                                                                setcolor(DARKGRAY);
 puthz(220,180,"出库完成",48,48,RED);
                                                                //setlinestyle(0,0,3);
 delay(1000);
                                                                setfillstyle(0,WHITE);
 // draw_home01();
                                                                for(i=0;i<4;i++)
 mode1=0;
                                                                {
}
                                                                     rectangle(x,d_y,x+50,d_y+30);
                                                                     bar(x,d y,x+50,d y+30);
//draw the trunk int the page
                                                                     d_y+=30;
                                                               }
void draw_trunk()
                                                              }
{
 setfillstyle(1,RED);
 //setcolor(DARKGRAY);
                                                               /*int main()
 bar(50,210,160,260);
 setfillstyle(1,LIGHTGRAY);
                                                                int gd=VGA,gm=VGAHI;
 bar(160,225,190,260);
                                                                initgraph(&gd,&gm,"..\borlandc\\bgi");
 setfillstyle(1,DARKGRAY);
                                                                draw home01();
 fillellipse(175,255,11,11);
                                                                closegraph();
 fillellipse(76,255,11,11);
                                                                return 0;
 fillellipse(104,255,11,11);
                                                         }*/
 setfillstyle(1,LIGHTBLUE);
                                                          LOGFUN.C
                                                 5、
 bar(165,230,185,240);
}
                                                               #include "COMMON.H"
                                                               #include "LOGFUN.H"
//draw the picture of warehouse
                                                               #include "PARAMETE.H"
void draw_warehouse()
                                                              // 录入注册账号到文件
 int i,j,y_d=160;
                                                              void wr_user(char
                                                                                      username1[],
                                                                                                       char
                                                          password1[], char phonenumber1[])
 setcolor(DARKGRAY);
 setlinestyle(0,0,3);
                                                              {
 line(0,230,640,230);
                                                                FILE *fp;
 for(i=0;i<4;i++)
                                                                int i,j;
                                                                user *u = (user *)malloc(sizeof(user));
      int x_d=400;
                                                                u->lenpar = 0;
      for(j=0;j<3;j++)
                                                                for (i = 0; i < 5; i++)
      {
           int m=rand()%10;
                                                                     u->here[i].cotton_type = 0;
           cot_mount(x_d+m,y_d);
                                                                for (i=0;i<5;i++)
           x d+=70;
      y_d+=80;
                                                                     u->here[i].ware_name[0]='\0';
 }
                                                                    for (j=0;j<3;j++)
}
                                                                    {
                                                                         u->here[i].total[j]=0;
//draw the cotton in warehouse
                                                                    }
```

```
}
                                                                   fseek(fp, 0, SEEK_END);
      if ((fp = fopen("User.dat", "rb+")) == NULL)
                                                                   all = ftell(fp) / sizeof(user);
                                                                   for (i = 0; i < all; i++)
          puthz(120, 300, "打开错误", 32, 32,
BLUE);
                                                                        flag = 0;
          delay(3000);
                                                                        fseek(fp, i * sizeof(user), SEEK SET);
                                                                        fread(u, sizeof(user), 1, fp);
          return;
     }
     for (i = 0; i < 10; i++)
                                                                        for (j = 0; j < 10; j++) // 查找账号位置
                                                                        {
      {
          u->username[i] = username1[i];
                                                                                     (u->username[j]
                                                                             if
                                                                                                            !=
          u->password[i] = password1[i];
                                                             username0[j])
     }
                                                                             {
     for (i = 0; i < 12; i++)
                                                                                  break;
          u->phonenumber[i]
                                                                             if (username0[j] == '\0')
phonenumber1[i];
                                                                             {
                                                                                 j = 10;
      fseek(fp, 0, SEEK END);
                                                                                 break;
     fwrite(u, sizeof(user), 1, fp);
                                                                             }
     free(u);
                                                                        }
      u = NULL;
                                                                        if (j == 10)
      if (fclose(fp)!=0) // 关闭文件
                                                                             setfillstyle(1, CYAN);
          puthz(120, 300, "关闭错误", 32, 32,
                                                                             bar(200, 30, 440, 72);
BLUE);
                                                                            puthz(210, 32, "注册账号已存在",
          delay(3000);
                                                             32, 32, BLUE);
          return;
                                                                             delay(1200);
     }
                                                                             setfillstyle(1, 0);
    }
                                                                             bar(200, 30, 440, 72);
                                                                            puthz(260, 30, "注册账号", 32, 32,
    // 判断注册的账号是否已经存在和电话号
                                                             BLUE);
码是否正确
                                                                             flag = 1;
    // return flag 为 1 则存在,0 则不存在
                                                                        }
    int username same(char username0[], char
                                                                   }
phonenumber0[])
                                                                   if (strlen(phonenumber0) != 11)
    {
                                                                   {
      FILE *fp;
                                                                        setfillstyle(1, CYAN);
      user *u = (user *)malloc(sizeof(user));
                                                                        bar(180, 30, 460, 72);
     int i, j, all, flag = 0;
                                                                        puthz(190, 32, "请输入十一位号码",
      if ((fp = fopen("User.dat", "rb+")) == NULL)
                                                             32, 32, BLUE);
                                                                        delay(1200);
          puthz(120, 300, "打开错误", 32, 32,
                                                                        setfillstyle(1, 0);
BLUE);
                                                                        bar(180, 30, 460, 72);
          delay(3000);
                                                                        puthz(260, 30, "注册账号", 32, 32,
          return 0;
                                                             BLUE);
     }
                                                                        flag = 1;
```

```
}
                                                                         fseek(fp, i * sizeof(user), SEEK_SET);
      if (flag == 0)
                                                                         fread(u, sizeof(user), 1, fp);
           setfillstyle(1, CYAN);
                                                                         for (j = 0; j < 10; j++) // 查找账号位置
           bar(200, 30, 440, 72);
                                                                                       (u->username[j]
          puthz(210, 32, "注册账号成功", 32, 32,
                                                                              if
                                                                                                             !=
BLUE);
                                                              username0[j])
           delay(1200);
                                                                              {
           setfillstyle(1, 0);
                                                                                   break;
           bar(200, 30, 440, 72);
           puthz(260, 30, "注册账号", 32, 32,
                                                                              if (username0[j] == '\0')
BLUE);
      }
                                                                                   j = 10;
      if (fclose(fp)!= 0) // 关闭文件
                                                                                   break;
           puthz(120, 300, "关闭错误", 32, 32,
                                                                         }
BLUE);
                                                                         if (j == 10) // 如果找到了账号就开始
           delay(3000);
                                                              比对密码
           return 0;
                                                                         {
      }
                                                                              for (k = 0; k < 10; k++)
      free(u);
                                                                              {
      u = NULL;
                                                                                   if
                                                                                         (u->password[k]
                                                                                                             1=
      return flag;
                                                              password0[k])
    }
                                                                                   {
                                                                                        break;
    // 登录,判断账号是否存在 and 密码是否
正确
                                                                                   if (password0[k] == '\0')
    int logg(char username0[], char password0[])
                                                                                        k = 10;
      FILE *fp;
                                                                                        break;
      user *u = (user *)malloc(sizeof(user));
                                                                                   }
      int i, j, k, l, o, flag = 0, all;
                                                                              }
      // char a[2];
                                                                              if (k == 10)
      // a[2] = '\0';
                                                                              {
                                                                                   strcpy(h->username,
      if ((fp = fopen("User.dat", "rb+")) == NULL)
                                                              u->username);
                                                                                   strcpy(h->password,
      {
           puthz(120, 300, "打开错误", 32, 32,
                                                              u->password);
BLUE);
                                                                                   strcpy(h->phonenumber,
           delay(3000);
                                                              u->phonenumber);
           return 0;
                                                                                   h->lenpar = u->lenpar;
      }
                                                                                   for (I = 0; I < u->lenpar; I++)
      fseek(fp, 0, SEEK_END);
                                                                                   {
      all = ftell(fp) / sizeof(user);
      for (i = 0; i < all; i++)
                                                               parcpy(&(h->parameter[I]),&(u->parameter[I]));
                                                                                   for (I = 0; I < 5; I++)
```

```
{
                                                                     else if (flag == 0)
                         // for (k = 0; k < 15; k++)
                         //{
                                                                          setfillstyle(1, CYAN);
                                                                          bar(240, 30, 400, 80);
 h->here[l].ware name[k]
                                                                          puthz(250, 35, "登录失败", 32, 32,
                                                              BLUE);
u->here[l].ware_name[k];
                         //}
                                                                          delay(1000);
                                                                          setfillstyle(1, 0);
 strcpy(h->here[I].ware_name,u->here[I].ware_na
                                                                          bar(240, 30, 400, 80);
                                                                          puthz(180, 30, "棉花模拟采集系统",
me);
                                                              32, 32, BLUE);
                         for (k = 0; k < 3; k++)
                              h->here[l].total[k]
                                                                     if (fclose(fp)!= 0) // 关闭文件
= u->here[l].total[k];
                                                                          puthz(120, 300, "关闭错误", 32, 32,
                         h->here[l].cotton_type
                                                              BLUE);
                                                                          delay(3000);
= u->here[l].cotton_type;
                                                                          return 0;
                                                                     }
                    flag = 1;
                                                                     free(u);
                    break;
               }
                                                                     u = NULL;
          }
                                                                     return flag;
      }
      if ((j != 10) && (flag == 0))
                                                                    int changepassword(char username0[], char
                                                              newpassword0[], char phonenumber0[])
           setfillstyle(1, CYAN);
           bar(220, 30, 420, 80);
                                                                   {
                                                                     FILE *fp;
           puthz(250, 35, "账号未注册", 32, 32,
BLUE);
                                                                     user *u = (user *)malloc(sizeof(user));
           delay(1000);
                                                                     int i, j, k, flag = 0, all;
           setfillstyle(1, 0);
           bar(220, 30, 420, 80);
                                                                     if ((fp = fopen("User.dat", "rb+")) == NULL)
           puthz(180, 30, "棉花模拟采集系统",
                                                                     {
32, 32, BLUE);
                                                                          puthz(120, 300, "打开错误", 32, 32,
                                                              BLUE);
      }
                                                                          delay(3000);
      else if ((k != 10) && (flag == 0))
      {
                                                                          return 0;
           setfillstyle(1, CYAN);
           bar(220, 30, 420, 80);
                                                                     fseek(fp, 0, SEEK_END);
                                                                     all = ftell(fp) / sizeof(user);
           puthz(250, 35, "密码不正确", 32, 32,
                                                                     for (i = 0; i < all; i++)
BLUE);
           delay(1000);
           setfillstyle(1, 0);
                                                                          fseek(fp, i * sizeof(user), SEEK SET);
           bar(220, 30, 420, 80);
                                                                         fread(u, sizeof(user), 1, fp);
           puthz(180, 30, "棉花模拟采集系统",
                                                                         for (j = 0; j < 10; j++) // 查找账号位置
32, 32, BLUE);
      }
```

```
if
                       (u->username[j]
                                             !=
username0[j])
                                                                 if (flag == 1)
               {
                   break;
                                                                      setfillstyle(1, CYAN);
                                                                      bar(220, 100, 420, 150);
               if (username0[j] == '\0')
                                                                      puthz(220, 100, "更改密码成功", 32,
                                                           32, BLUE);
                   j = 10;
                                                                      delay(1500);
                   break;
                                                                 }
                                                                 else
               }
          }
                                                                 {
          if (j == 10) // 找到了就判断电话号码
                                                                      setfillstyle(1, CYAN);
正不正确
                                                                      bar(220, 80, 420, 130);
                                                                     puthz(220, 85, "更改密码失败", 32, 32,
          {
                                                           BLUE);
              for (j = 0; j < 12; j++)
                                                                      delay(1200);
                                                                      setfillstyle(1, LIGHTBLUE);
               {
                                                                      bar(220, 80, 420, 130);
                                                                 }
                       (u->phonenumber[j] !=
phonenumber0[j])
                   {
                                                                 free(u);
                        //
                                                                 u = NULL;
outtextxy(200,200,u->phonenumber);
                                                                 if (fclose(fp)!= 0) // 关闭文件
                        break;
                   }
                                                                      puthz(120, 300, "关闭错误", 32, 32,
                   if (phonenumber0[j] == '\0')
                                                           BLUE);
                                                                      delay(3000);
                   {
                        j = 12;
                                                                      return 0;
                        break;
                                                                 }
                   }
                                                                 return flag;
                                                           }
               if (j == 12) // 电话号码正确就改
                                                   6
                                                           LOGIN.C
密码
                                                                #include "COMMON.H"
                   //
                                                                #include "PARAMETE.H"
outtextxy(200,200,u->phonenumber);
                                                                #include "LOGFUN.H"
                                                                #include "LOGIN.H"
                   for (k = 0; k < 10; k++)
                                                                #include "REGISTER.H"
                        u->password[k]
                                                                #include "RESET.H"
newpassword0[k];
                   fseek(fp, i * sizeof(user),
                                                                void loginit screen()
SEEK_SET);
                                                                {
                   fwrite(u, sizeof(user), 1, fp);
                                                                     clrmous(MouseX,MouseY);
                   flag = 1;
                                                                    cleardevice();
                                                                     setbkcolor(WHITE);
              }
          }
                                                                     puthz(180,30," 棉花模拟采集系统
```

```
",32,32,BLUE);
                                                                       INPUT username = {220, 100, 520,
                                                             160,"",10,0,0};
         quit();
                                                                   INPUT password = {220,
                                                                                                   200.
                                                                                                          520,
                                                             260,"",10,0,0};
         setcolor(BLUE);
                                                                       loginit screen();
         setfillstyle(1,LIGHTGRAY);
                                                                       clrmous(MouseX, MouseY);
         //rectangle(123, 103, 523, 153);
                                                                       for(;;)
         bar(120, 100, 220, 160);
                                                                   {
         //rectangle(123, 203, 523, 253);
         bar(120, 200, 220, 260);
                                                               newmouse(&MouseX,&MouseY,&press);
         setfillstyle(1,LIGHTGREEN);
                                                                           if(logenter()==1)
         rectangle(220, 100, 520, 160);
         rectangle(220, 200, 520, 260);
         //bar(220, 100, 520, 160);
                                                             if(logg(username.string,password.string)==1)
         //bar(220, 200, 520, 260);
                                                                                     return;
         bar(280, 300, 360, 340);
                                                                                }
         setfillstyle(1,DARKGRAY);
                                                                           input_s(223, 100, &username, 16,
         bar(120, 350, 280, 400);
                                                             0);
         bar(360, 350, 520, 400);
                                                                           input s(223, 200, &password, 16,
                                                             1);
         puthz(123,110,"账号: ",32,32,BLUE);
                                                                           if(mouse_press(360,
                                                                                                  350,
                                                                                                          520,
         puthz(123,210,"密码: ",32,32,BLUE);
                                                             400)==1)
                            忘
         puthz(130,360,"
                                  记
                                              码
                                                                           {
",32,32,WHITE);
                                                                                log register();
         puthz(370,360,"
                            注
                                  册
                                        账
                                              号
                                                                                loginit_screen();
",32,32,WHITE);
                                                                           if(mouse_press(120,
         puthz(285,305,"登录",32,32,BLUE);
                                                                                                  350,
                                                                                                          280,
    }
                                                             400) = = 1)
                                                                           {
    int logenter()
                                                                                reset();
                                                                                loginit_screen();
         if(mouse\_press(0,0,40,30)==1)
                                                                           }
     {
          exit(0);
                                                                   delay(15);
     }
         if(mouse_press(280, 300, 360, 340)==1)
                                                             }
         {
                                                     7、
                                                             MAIN.C
              return 1;
                                                                  #include "COMMON.H"
         }
         return 0;
                                                                  #include "LOGFUN.H"
    }
                                                                  #include "LOGIN.H"
                                                                  #include "PARAMETE.H"
                                                                  #include "START.H"
    void loginit()
    {
                                                                  #include "HOME.H"
```

```
#include "HELP.H"
                                                                                             mode1=0;
    #include "EDIT.H"
                                                                                             break;
    #include "WELCOME.H"
                                                                                         }
    #include "PAST.H"
                                                                                         case 0: //the home
    struct User *h;//登录的用户
                                                             main page
    int mode=0;
                                                                                         {
    int mode1=0;
                                                             draw_home01();
    void main()
                                                                                             break;
                                                                                         }
         int gd=VGA,gm=VGAHI,i=0;
                                                                                                        //the
                                                                                         case
                                                                                                 1:
      h=(user*)malloc(sizeof(user));// 登录的用
                                                             warehouse list page
户
                                                                                         {
      initgraph(&gd,&gm,"..\\borlandc\\bgi");
         loginit();
                                                             warehouse_list(h->here);
         for(i=0;i<5;i++)
                                                                                             break;
                                                                                         }
         {
                                                                                                 2:
                                                                                                        //the
                                                                                         case
if(strcmp(h->here[i].ware_name,"\0")==0)
                                                             detailed warehouse page
                                                                                         {
                  char s[10]="ware0";
                  s[5]='0'+i+1;
                                                             detailed_warehouse(h->here[k].total[(h->here[k].
                                                             cotton_type)]);
strcpy(h->here[i].ware_name,s);
                                                                                             break;
                                                                                         }
              }
         }
                                                                                         case 3: //the export
         while(1)
                                                             cotton page
                                                                                         {
              int
pre_mode=mode,pre_mode1=mode1;
                                                             out_warehouse();
              switch (mode)
                                                                                             break;
                                                                                         }
                  case 0: //the welcome page
                                                                                    }
                  {
                                                                                    break;
                       draw_wel();
                                                                               }
                       break;
                                                                                case 2: //the edit page
                  }
                                                                               {
                                                                                    edit();
                  case 1: //the home page
                                                                                    pre_mode=2;
                  {
                                                                                    mode=2;
                                                                                    break;
                       switch(mode1)
                                                                               }
                       {
                            case -1:
                                                                               case 3: // the start page
                            {
                                //
                                                                                    start();
draw_home01();
                                                                                    pre_mode=3;
```

```
// mode=3;
                                                                                                 break;
                       break;
                                                                                             }
                  }
                                                                                             case 0: //the
                                                            home main page
                  case 4: //the past page
                                                                                             {
                                                            press_home(&(h->here[k].cotton_type));
                       past();
                       pre_mode=4;
                                                                                                 break;
                       mode=4;
                                                                                             }
                       break;
                                                                                             case 1: //the
                  }
                                                            warehouse list page
                                                                                             {
                  case 5: //the help page
                                                                                                 int re=0;
                       help();
                                                            press_warelist(&re);
                       break;
                                                                                                 break;
                  }
                                                                                             }
                                                                                             case 2: //the
                  default:
                                                            detailed warehouse page
                       break;
                                                                                             {
              }
                                                            press_detwarehouse();
while(pre_mode==mode&&pre_mode1==mode1)
                                                                                                 break;
              {
                                                                                             }
                                                                                             case 3: //the
newmouse(&MouseX,&MouseY,&press);
                                                            export cotton page
                  switch (mode)
                                                                                             {
                       case 0: //the welcome
                                                            press_outware();
page
                                                                                                  break;
                       {
                                                                                             }
                            enter_next();
                                                                                        }
                            break;
                                                                                        break;
                       }
                                                                                   case 2: //the edit page
                       case 1: //the home page
                                                                                   {
                       {
                                                                                        cleardevice();
                            switch(mode1)
                                                                                        mode=0;
                            {
                                                                                        break;
                                             -1:
                                                                                   }
                                case
//refresh the home main page
                                                                                   case 3: // the start page
                                {
                                                                                   {
clrmous(MouseX,MouseY);
                                                                                        cleardevice();
                                                                                        mode=0;
                                     //
press_home(&(here[k].cotton_type));
                                                                                        break;
                                                                                   }
                                     mode1=0;
```

```
p = 1;
                       case 4: //the past page
                                                                     }
                       {
                                                                     if (press == 1)
                            cleardevice();
                                                                                    (mouse press(word->x1,
                            mode=0;
                                                                          if
                            break;
                                                            word->y1, word->x2, word->y2) == 1)
                       }
                                                                          {
                                                                               word->flag = 1;
                       case 5: //the help page
                       {
                                                                               clrmous(MouseX, MouseY);
                            cleardevice();
                                                                               setcolor(RED);
                            mode=0;
                                                                               setlinestyle(0, 0, 1);
                            break;
                                                                               rectangle(word->x1, word->y1,
                       }
                                                            word->x2, word->y2);
                       default:
                                                                               setcolor(DARKGRAY);
                            break;
                                                                               k = 1;
                                                                          }
                  }
                  delay(20);
                                                                          else
             }
                                                                          {
         }
                                                                               word - flag = 0;
    }
                                                                               clrmous(MouseX, MouseY);
                                                                               setcolor(BLUE);
PARAMETE.C
                                                                               setlinestyle(0, 0, 1);
    include "COMMON.H"
                                                                               rectangle(word->x1, word->y1,
    #include "PARAMETE.H"
                                                            word->x2, word->y2);
    #include "LOGFUN.H"
                                                                               // 不可输入则把光标遮蔽掉
    #include "START.H"
                                                                               if (mode == 0)
    // 专门用来写有关 edit 和 past 的函数
                                                                               {
                                                                                   setfillstyle(1, 0);
    // 键盘输入
                    mode 为 0 输出文字, mode
                                                                                   bar(word->x1
                                                                                                          2,
为1输出*
                                                            word->y1 + 2, word->x2 - 2, word->y2 - 2);
    int input_s(int x, int y, INPUT *word, int size,
                                                                                   setcolor(DARKGRAY);
int mode)
                                                                                   outtextxy(x,
                                                                                                          у,
    {
                                                            word->string);
         static int p = 0; // 画一个框
                                                                               }
                            // 判断是否输出文
         int k = 0;
                                                                               else
                                                                               {
         settextjustify(0, 2);
                                                                                   int i;
         if (p == 0)
                                                                                   setfillstyle(1, 0);
                                                                                                          2,
                                                                                   bar(word->x1
         {
              setcolor(BLUE);
                                                            word->y1 + 2, word->x2 - 2, word->y2 - 2);
              rectangle(word->x1,
                                      word->v1,
                                                                                   for
                                                                                        (i = 0;
word->x2, word->y2);
                                                            word->cursor; i++)
             setfillstyle(1, 0);
                                                                                   {
              bar(word->x1 + 2, word->y1 + 2,
                                                                                        outtextxy(x + i * (2 *
word->x2 - 2, word->y2 - 2);
                                                            size - 2), y, "*");
```

8

字

```
setfillstyle(SOLID_FILL,
                         setcolor(DARKGRAY);
                                                                 WHITE);
                    }
                                                                                     settextjustify(LEFT TEXT,
                                                                 TOP_TEXT);
               }
         }
                                                                                      settextstyle(SMALL FONT,
                                                                 HORIZ_DIR, size);
          // flag 为 1 时表示可以接收键盘输入
                                                                                      bar(word->x1 + 2, word->y1 +
          if (word->flag == 1)
                                                                 2, word->x2 - 2, word->y2 - 2);
          {
                                                                                     if (mode == 0)
                                                                                      {
               char t;
                                                                                          setcolor(DARKGRAY);
               if (kbhit())
                                                                                          outtextxy(x,
                                                                                                                   у,
               {
                                                                 word->string);
                    t = getch();
                                                                                          line(x + (word->cursor) *
                                                                 (2 * size - 8) + 2, word->y1 + 3, x + (word->cursor)
                    if (t == '\b')
                                                                 * (2 * size - 8) + 2, word->y2 - 3);
                                                                                     }
                         if (word->cursor > 0)
                                                                                     else
                                                                                     {
                                                                                          int i;
(word->string)[word->cursor - 1] = '\0';
                                                                                          for
                                                                                                 (i
                                                                                                          0;
                              (word->cursor)--;
                                                                 word->cursor; i++)
                              k = 1;
                                                                                          {
                         }
                                                                                                outtextxy(x + i * (2 *
                    }
                                                                 size - 2), y, "*");
                    else if (t >= '!' \&\& t <= '\sim')
                                                                                          }
                    {
                                                                                          setcolor(DARKGRAY);
                         if
                               (word->cursor
                                                                                          line(x + (word->cursor) *
                                                                 (2 * size - 2) + 2, word->y1 + 3, x + (word->cursor)
word->length)
                         {
                                                                 * (2 * size - 2) + 2, word->y2 - 3);
                                                                                     }
(word->string)[word->cursor] = t;
                                                                           }
(word->string)[word->cursor + 1] = '\0';
                                                                           return 0;
                              (word->cursor)++;
                                                                      }
                              k = 1;
                         }
                                                                      int judgename(char name[])
                         else
                         {
                                                                           static int i, j;
                              return 1;
                                                                           if (name[0] == '\0')
                         }
                    }
                                                                                void *buffer;
                                                                                unsigned s;
               }
               if (k == 1)
                                                                                setfillstyle(1, CYAN);
                                                                                bar(240, 30, 470, 70);
                    setcolor(DARKGRAY);
                                                                                puthz(242, 33, "参数名不能为空",
                    setlinestyle(0, 0, 1);
                                                                 32, 32, BLUE);
```

```
puthz(83, 100, "土地面积不能为
              delay(1000);
              setfillstyle(1, 0);
                                                              空", 32, 32, BLUE);
              bar(240, 30, 470, 70);
                                                                            delay(1000);
              puthz(240, 30, "请选择地区", 32,
                                                                            setfillstyle(1, 0);
32, BLUE);
                                                                            bar(80, 100, 380, 140);
                                                                            puthz(70, 100, "本地推荐种植的
              return 0;
                                                              棉花种类为: ", 32, 32, BLUE);
         }
         for (i = 0; i < h->lenpar; i++)
                                                                            return 0;
                                                                       }
              for (j = 0; j < 10; j++)
                                                                       for (i = 0; i < strlen(S); i++)
                             (name[i]
                   if
                                               !=
                                                                            if (S[i] > '9' | | S[i] < '0')
(h->parameter[i]).name[j])
                                                                            {
                   {
                                                                                 setfillstyle(1, CYAN);
                        break;
                                                                                 bar(80, 100, 340, 140);
                                                                                 puthz(83, 100, "请输入数字",
                   }
                   if (name[i] == '\0')
                                                              32, 32, BLUE);
                                                                                 delay(1000);
                       i = 10;
                                                                                 setfillstyle(1, 0);
                                                                                 bar(80, 100, 340, 140);
                        break;
                   }
                                                                                 puthz(70, 100, "本地推荐种
              }
                                                              植的棉花种类为: ", 32, 32, BLUE);
              if (j == 10)
                                                                            return 0;
              {
                                                                            }
                   setfillstyle(1, CYAN);
                                                                       }
                   bar(240, 30, 470, 70);
                                                                        if (S[0] == '0')
                   puthz(242, 33, "参数名已经
                                                                       {
存在", 32, 32, BLUE);
                                                                            setfillstyle(1, CYAN);
                   delay(1000);
                                                                            bar(80, 100, 380, 140);
                   setfillstyle(1, 0);
                                                                             puthz(83, 100, "土地面积不能为
                   bar(240, 30, 470, 70);
                                                              零", 32, 32, BLUE);
                   puthz(240, 30, "请选择地区",
                                                                            delay(1000);
32, 32, BLUE);
                                                                            setfillstyle(1, 0);
                                                                            bar(80, 100, 380, 140);
                                                                            puthz(70, 100, "本地推荐种植的
                   return 0;
                                                              棉花种类为: ", 32, 32, BLUE);
              }
         }
                                                                            return 0;
                                                                       }
         return 1;
    }
                                                                        return 1;
                                                                   }
    int judgeS(char S[])
                                                                   void wr parameter(struct Parameter *abc)
         int i = 0;
                                                                   {
         if (S[0] == '\0')
                                                                       FILE *fp;
                                                                        user *u = (user *)malloc(sizeof(user));
              setfillstyle(1, CYAN);
                                                                        int i, j, k, all;
              bar(80, 100, 380, 140);
                                                                        if ((fp = fopen("User.dat", "rb+")) ==
```

```
NULL)
         {
                                                            strcpy(h->parameter[h->lenpar].S, abc->S);
              puthz(120, 300, "打开错误", 32, 32,
                                                                                                0;
                                                                                                     k
                                                                                                           <
BLUE);
                                                            dense_points_max; k++)
              delay(3000);
              return;
         }
                                                            h->parameter[h->lenpar].x[k] = abc->x[k];
         fseek(fp, 0, SEEK_END);
         all = ftell(fp) / sizeof(user); // 文件里
                                                            h->parameter[h->lenpar].y[k] = abc->y[k];
user 的数量
         for (i = 0; i < all; i++)
                                                            h->parameter[h->lenpar].lenxy = abc->lenxy;
                                    sizeof(user),
                                                                               h->lenpar += 1;
              fseek(fp,
                                                                               fseek(fp, i *
SEEK_SET);
                                                                                                sizeof(user),
              fread(u, sizeof(user), 1, fp);
                                                            SEEK_SET);
              for (j = 0; j < 10; j++) // 查找账号
                                                                               fwrite(h, sizeof(user), 1, fp);
位置
                                                                               puthz(220, 100, "增加参数成
              {
                                                            功", 32, 32, BLUE);
                                                                               delay(2000);
                         (u->username[j]
                  if
                                             !=
                                                                          }
h->username[j])
                                                                     }
                                                                     // char name[10];//名字
                  {
                       break;
                                                                     // char place;//产区
                                                                     // char shape;//土地形状
                  if (h-susername[j] == '\0')
                                                                     // char type;//收割机类型
                  {
                                                                     // int S;//面积
                       j = 10;
                                                                     // int xyz[200];//坐标
                       break;
                  }
                                                                     free(u);
              }
                                                                      u = NULL;
              if (j == 10) // 找到了就修改参数
                                                                     if (fclose(fp)!=0) // 关闭文件
                  //
                                                                          puthz(120, 300, "关闭错误", 32, 32,
strcpy(h->parameter[h->lenpar].name,abc->name
                                                            BLUE);
);
                                                                          delay(3000);
                  for (k = 0; k < 10; k++)
                                                                          return;
                  {
                                                                     return;
h->parameter[h->lenpar].name[k]
                                                                 }
abc->name[k];
                  }
                                                                 // 把当前登录的用户重新写进文件
                                                                 void wr h(void)
h->parameter[h->lenpar].place = abc->place;
                                                                 {
                                                                     FILE *fp;
                                                                     user *u = (user *)malloc(sizeof(user));
h->parameter[h->lenpar].shape = abc->shape;
                  h->parameter[h->lenpar].type
                                                                     int i, j, k, all;
                                                                     if ((fp = fopen("User.dat", "rb+")) ==
= abc->type;
```

```
NULL)
                                                                  }
         {
              puthz(120, 300, "打开错误", 32, 32,
                                                                  // 删除参数
BLUE);
                                                                  void deletepar(int par)
              delay(3000);
                                                                  {
              return;
                                                                       int i, j, k;
         }
                                                                       par -= 1;
         fseek(fp, 0, SEEK_END);
                                                                       for (i = par; i < h->lenpar; i++)
         all = ftell(fp) / sizeof(user); // 文件里
user 的数量
                                                                            parcpy(&(h->parameter[par]),
         for (i = 0; i < all; i++)
                                                             &(h->parameter[par + 1]));
                                    sizeof(user),
                                                                       h->lenpar -= 1;
              fseek(fp,
                          i
SEEK_SET);
                                                                       wr_h();
              fread(u, sizeof(user), 1, fp);
                                                                  }
              for (j = 0; j < 10; j++)
                                                                  // 下面四个返回 1 为修改成功,返回-1 为
              {
                                                             无修改
                  if
                         (u->username[i]
                                              !=
h->username[j])
                                                                  // 修改产区
                  {
                                                                  int changeplace(int par)
                       break;
                                                                       char choose;
                  if (h-susername[j] == '\0')
                                                                       clrmous(MouseX, MouseY);
                                                                       setfillstyle(1, BROWN);
                                                                       bar(100, 130, 540, 370);
                       j = 10;
                       break;
                                                                       puthz(220, 150, "请重新选择产区", 32,
                  }
                                                             32, BLUE);
              }
                                                                       setfillstyle(1, CYAN);
              if (j == 10) // 找到了就把 h 传进去
                                                                       bar(150, 290, 240, 340);
                                                                       puthz(153, 293, "返回", 32, 32, BLUE);
              {
                   fseek(fp, i * sizeof(user),
                                                                       bar(400, 290, 490, 340);
SEEK_SET);
                                                                       puthz(403, 293, "确认", 32, 32, BLUE);
                   fwrite(h, sizeof(user), 1, fp);
              }
                                                                       setfillstyle(1, LIGHTRED);
         }
                                                                       bar(305, 240, 335, 270);
                                                                       puthz(300, 200, "长江", 16, 16, BLUE);
                                                                       bar(180, 240, 210, 270);
         free(u);
         u = NULL;
                                                                       puthz(175, 200, "新疆", 16, 16, BLUE);
         if (fclose(fp)!=0) // 关闭文件
                                                                       bar(430, 240, 460, 270);
                                                                       puthz(425, 200, "黄河", 16, 16, BLUE);
         {
              puthz(120, 300, "关闭错误", 32, 32,
BLUE);
                                                                       quit();
              delay(3000);
              return;
                                                                       for (;;)
                                                                       {
         }
                                                                            delay(15);
         return;
```

```
newmouse(&MouseX,
                                       &MouseY,
&press);
                                                                            if (mouse_press(400, 290, 490, 340)
                                                              == 1) // 确认
              if (mouse_press(305, 240, 335, 270)
                                                                            {
== 1) // 长江
                                                                                 h->parameter[par - 1].place =
              {
                                                              choose;
                   clrmous(MouseX, MouseY);
                                                                                 wr h();
                   choose = 'c';
                                                                                 return 1;
                   setfillstyle(1, BROWN);
                                                                            }
                   bar(180, 240, 460, 270);
                                                                       }
                   setfillstyle(1, BLUE);
                                                                  }
                   bar(305, 240, 335, 270);
                   setfillstyle(1, LIGHTRED);
                                                                  // 修改土地形状
                   bar(180, 240, 210, 270);
                                                                  int changeshape(int par)
                   bar(430, 240, 460, 270);
                                                                       char choose;
              }
              if (mouse press(180, 240, 210, 270)
                                                                       clrmous(MouseX, MouseY);
== 1) // 新疆
                                                                       setfillstyle(1, BROWN);
                                                                       bar(100, 130, 540, 370);
                   clrmous(MouseX, MouseY);
                                                                       puthz(200, 150, "请重新选择土地形状",
                   choose = 'a';
                                                              32, 32, BLUE);
                   setfillstyle(1, BROWN);
                                                                       setfillstyle(1, CYAN);
                   bar(180, 240, 460, 270);
                                                                       bar(150, 290, 240, 340);
                                                                       puthz(153, 293, "返回", 32, 32, BLUE);
                   setfillstyle(1, BLUE);
                   bar(180, 240, 210, 270);
                                                                       bar(400, 290, 490, 340);
                                                                       puthz(403, 293, "确认", 32, 32, BLUE);
                   setfillstyle(1, LIGHTRED);
                   bar(305, 240, 335, 270);
                   bar(430, 240, 460, 270);
                                                                       setfillstyle(1, YELLOW);
              }
                                                                       pieslice(170, 240, 0, 360, 20);
                                                                       puthz(155, 195, "矩形", 16, 16, BLUE);
              if (mouse_press(430, 240, 460, 270)
== 1) // 黄河
                                                                       pieslice(270, 240, 0, 360, 20);
                                                                       puthz(255, 195, "圆形", 16, 16, BLUE);
              {
                                                                       pieslice(370, 240, 0, 360, 20);
                   clrmous(MouseX, MouseY);
                   choose = 'b';
                                                                       puthz(340, 195, "多边形", 16, 16,
                   setfillstyle(1, BROWN);
                                                              BLUE);
                                                                       pieslice(470, 240, 0, 360, 20);
                   bar(180, 240, 460, 270);
                   setfillstyle(1, BLUE);
                                                                       puthz(425, 195, "自定义图形", 16, 16,
                   bar(430, 240, 460, 270);
                                                              BLUE);
                   setfillstyle(1, LIGHTRED);
                   bar(180, 240, 210, 270);
                                                                       for (;;)
                   bar(305, 240, 335, 270);
              }
                                                                            delay(15);
                                                                            newmouse(&MouseX,
                                                                                                     &MouseY,
              if (mouse_press(150, 290, 240, 340)
                                                              &press);
== 1) // 返回
                                                                            if (mouse_press(150, 220, 190, 360)
              {
                                                              == 1) // 矩形
                                                                            {
                   return -1;
```

```
clrmous(MouseX, MouseY);
                                                                                    pieslice(270, 240, 0, 360, 20);
                   choose = 'a';
                                                                                    pieslice(370, 240, 0, 360, 20);
                   setfillstyle(1, BROWN);
                                                                                    pieslice(170, 240, 0, 360, 20);
                   bar(150, 220, 490, 260);
                                                                                    setfillstyle(1, BLUE);
                   setfillstyle(1, YELLOW);
                                                                                    pieslice(470, 240, 0, 360, 20);
                                                                               }
                   pieslice(270, 240, 0, 360, 20);
                   pieslice(370, 240, 0, 360, 20);
                                                                               if (mouse press(150, 290, 240, 340)
                                                                == 1) // 返回
                   pieslice(470, 240, 0, 360, 20);
                    setfillstyle(1, BLUE);
                                                                               {
                   pieslice(170, 240, 0, 360, 20);
                                                                                    return -1;
              }
              if (mouse press(250, 220, 290, 360)
                                                                               if (mouse press(400, 290, 490, 340)
== 1) // 圆形
                                                                == 1) // 确认
              {
                                                                               {
                   clrmous(MouseX, MouseY);
                                                                                    if (choose == 'c')
                   choose = 'b';
                   setfillstyle(1, BROWN);
                   bar(150, 220, 490, 260);
                                                                select02(&(h->parameter[par - 1]));
                   setfillstyle(1, YELLOW);
                                                                                    else if (choose == 'd')
                   pieslice(170, 240, 0, 360, 20);
                   pieslice(370, 240, 0, 360, 20);
                                                                                    {
                   pieslice(470, 240, 0, 360, 20);
                    setfillstyle(1, BLUE);
                                                                select03(&(h->parameter[par - 1]));
                    pieslice(270, 240, 0, 360, 20);
              }
                                                                                    h->parameter[par - 1].shape =
               if (mouse press(350, 220, 390, 360)
                                                                choose;
== 1) // 多边形
                                                                                    wr_h();
                                                                                    return 1;
                   clrmous(MouseX, MouseY);
                                                                               }
                    choose = 'c';
                                                                          }
                   setfillstyle(1, BROWN);
                                                                     }
                   bar(150, 220, 490, 260);
                                                                     // 修改收割机类型
                   setfillstyle(1, YELLOW);
                   pieslice(270, 240, 0, 360, 20);
                                                                     int changetype(int par)
                    pieslice(170, 240, 0, 360, 20);
                   pieslice(470, 240, 0, 360, 20);
                                                                          char choose;
                    setfillstyle(1, BLUE);
                                                                          clrmous(MouseX, MouseY);
                   pieslice(370, 240, 0, 360, 20);
                                                                          setfillstyle(1, BROWN);
              }
                                                                          bar(100, 130, 540, 370);
              if (mouse press(450, 220, 490, 360)
                                                                          puthz(220, 150, "请重新选择产区", 32,
== 1) // 自定义图形
                                                                32, BLUE);
              {
                                                                          setfillstyle(1, CYAN);
                   clrmous(MouseX, MouseY);
                                                                          bar(150, 290, 240, 340);
                   choose = 'd';
                                                                          puthz(153, 293, "返回", 32, 32, BLUE);
                   setfillstyle(1, BROWN);
                                                                          bar(400, 290, 490, 340);
                   bar(150, 220, 490, 260);
                                                                          puthz(403, 293, "确认", 32, 32, BLUE);
                   setfillstyle(1, YELLOW);
```

```
setfillstyle(1, YELLOW);
                                                                                  wr_h();
         pieslice(200, 240, 0, 360, 20);
                                                                                  return 1;
         puthz(175, 195, "垂直式", 16, 16,
                                                                             }
BLUE);
                                                                        }
         pieslice(440, 240, 0, 360, 20);
                                                                   }
         puthz(415, 195, "水平式", 16, 16,
BLUE);
                                                                   // 修改面积
                                                                   int changeS(int par)
         for (;;)
                                                                   {
         {
                                                                        INPUT S = {245, 220, 445, 260, "", 6, 0,
              delay(15);
                                                              0};
              newmouse(&MouseX,
                                                                        clrmous(MouseX, MouseY);
                                       &MouseY,
&press);
                                                                        setfillstyle(1, BROWN);
              if (mouse press(180, 220, 220, 260)
                                                                        bar(100, 130, 540, 370);
== 1) // 垂直式
                                                                        puthz(220, 150, "请重新输入面积", 32,
                                                              32, BLUE);
              {
                   clrmous(MouseX, MouseY);
                                                                        setfillstyle(1, CYAN);
                   choose = 'a';
                                                                        bar(150, 290, 240, 340);
                                                                        puthz(153, 293, "返回", 32, 32, BLUE);
                   setfillstyle(1, BROWN);
                   bar(180, 220, 460, 260);
                                                                        bar(400, 290, 490, 340);
                   setfillstyle(1, YELLOW);
                                                                        puthz(403, 293, "确认", 32, 32, BLUE);
                   pieslice(440, 240, 0, 360, 20);
                                                                        puthz(170, 220, "面积: ", 32, 32, BLUE);
                   setfillstyle(1, BLUE);
                                                                        setfillstyle(1, 0);
                   pieslice(200, 240, 0, 360, 20);
                                                                        bar(245, 220, 445, 260);
              }
              if (mouse press(420, 220, 460, 260)
                                                                        for (;;)
== 1) // 水平式
                                                                        {
                                                                             delay(15);
                   clrmous(MouseX, MouseY);
                                                                             newmouse(&MouseX,
                                                                                                      &MouseY,
                   choose = 'b';
                                                              &press);
                   setfillstyle(1, BROWN);
                                                                             input s(248, 220, &S, 16, 0);
                   bar(180, 220, 460, 260);
                   setfillstyle(1, YELLOW);
                                                                             if (mouse_press(150, 290, 240, 340)
                   pieslice(200, 240, 0, 360, 20);
                                                              == 1) // 返回
                   setfillstyle(1, BLUE);
                                                                             {
                   pieslice(440, 240, 0, 360, 20);
                                                                                  return -1;
              }
              if (mouse press(150, 290, 240, 340)
                                                                             if (mouse press(400, 290, 490, 340)
== 1) // 返回
                                                              == 1) // 确认
                                                                             {
                   return -1;
                                                                                  if (judgeS(S.string) == 1)
              if (mouse press(400, 290, 490, 340)
                                                                                      strcpy(h->parameter[par
== 1) // 确认
                                                              - 1].S, S.string);
                                                                                      wr_h();
                   h->parameter[par - 1].type =
                                                                                  }
choose;
                                                                                  return 1;
```

```
}
                                                                            {
          }
                                                                                  char a[1];
    }
                                                                                  a[1] = '\0';
                                                                                  if ((h->lenpar / 4) == i)
    void parcpy(struct Parameter *a, struct
Parameter *b)
                                                                                       itoa(i, a, 10);
    {
                                                                                       page[2] = a[0];
          int k;
                                                                                       if ((h->lenpar % 4) != 0)
          for (k = 0; k < 10; k++)
          {
                                                                                            itoa(i + 1, a, 10);
               a->name[k] = b->name[k];
                                                                                            page[2] = a[0];
                                                                                       }
          }
          a->place = b->place;
                                                                                  }
          a->shape = b->shape;
                                                                            }
          a->type = b->type;
                                                                            for (;;)
          strcpy(a->S, b->S);
                                                                            {
          for (k = 0; k < dense_points_max; k++)
                                                                                  int pagepar;
                                                                                  char page2[1];
          {
               a->x[k] = b->x[k];
                                                                                  page2[1] = '\0';
               a->y[k] = b->y[k];
                                                                                  page2[0] = page[0];
          }
                                                                                  pagepar = atoi(page2) - 1;
          a->lenxy = b->lenxy;
                                                                                  newmouse(&MouseX,
                                                                                                            &MouseY,
    }
                                                                  &press);
    int choosepar(void)
                                                                                  if (flag == 1)
    {
                                                                                  {
          int i, j, flag = 1;
                                                                                       if (h->lenpar == 0)
          char page[3] = {'1', '/', '1'};
          int barcolor[11] = {1, 2, 3, 4, 5, 9, 10, 11,
                                                                                            setfillstyle(1,
12, 13, 14};
                                                                  LIGHTGRAY);
          // delay(400);
                                                                                            bar(50, 80, 590, 420);
          clrmous(MouseX, MouseY);
                                                                                            settextjustify(1, 1);
          cleardevice();
                                                                                            settextstyle(0, 0, 4);
          setbkcolor(WHITE);
                                                                                            setcolor(RED);
          puthz(240, 30, "请选择参数", 32, 32,
                                                                                            outtextxy(320, 250, "No
BLUE);
                                                                  Parameter");
          quit();
                                                                                            flag = 0;
          last();
                                                                                       }
          delay(15);
                                                                                       else
                                                                                       {
          setfillstyle(1, LIGHTGRAY);
                                                                                            setfillstyle(1,
          bar(50, 80, 590, 420);
                                                                  LIGHTGRAY);
                                                                                            bar(50, 80, 590, 420);
          settextstyle(0, 0, 2);
          settextjustify(1, 1);
                                                                                            settextstyle(0, 0, 2);
                                                                                            outtextxy(320,
                                                                                                                  405,
          page[3] = '\0';
                                                                  page);
          for (i = 0; i < 10; i++)
                                                                                                                  405,
                                                                                            outtextxy(280,
```

```
"<<");
                                                                                          flag = 0;
                         outtextxy(360,
                                               405,
">>");
                                                                                     }
                         settextstyle(0, 0, 3);
                                                                                }
                                                                                if (mouse press(260, 395, 310, 415)
                         setcolor(DARKGRAY);
                                                                 == 1)
                         for (i = 0, j = pagepar * 4;
                                                                                {
i < (4 > ((h->lenpar) - (pagepar * 4)) ? ((h->lenpar))
                                                                                     if (page[0] > '1')
- (pagepar * 4)) : 4); i++, j++)
                         {
                                                                                          page[0] -= 1;
                                                                                          flag = 1;
                              int a, b;
                              a = barcolor[rand() %
                                                                                          delay(100);
12];
                                                                                     }
                              b = barcolor[rand() %
                                                                                     else
12];
                                                                                     {
                              if (a == b)
                                                                                          setfillstyle(1, CYAN);
                                                                                          bar(230, 30, 450, 80);
                                                                                          puthz(240, 30, "第一页
                                   a += 1;
                                                                 啦", 32, 32, BLUE);
                              setcolor(a);
                                                                                          delay(300);
                              setfillstyle(1, b);
                                                                                          setfillstyle(1, 0);
                              bar(70, 150 + 60 * i,
                                                                                          bar(230, 30, 450, 80);
320, 150 + 40 + 60 * i);
                                                                                          puthz(240, 30, "参数列
                                                                 表", 32, 32, BLUE);
(strlen(h->parameter[j].name) >= 7)
                                                                                     }
                          {
                                                                                }
                               settextstyle(0,
                                                 0,
                                                                                // page next
2);
                                                                                if (mouse_press(330, 395, 380, 415)
                          }
                                                                 == 1)
                          else
                                                                                {
                          {
                                                                                     if (page[0] < page[2])
                               settextstyle(0,
                                                 0,
3);
                                                                                          page[0] += 1;
                          }
                                                                                          flag = 1;
                              outtextxy(195, 170 +
                                                                                          delay(100);
60 * i, h->parameter[j].name);
                                                                                     }
                         }
                                                                                     else if (page[0] == page[2])
                         for (i = 0; i < (4 >
                                                                                          setfillstyle(1, CYAN);
((h->lenpar) - (pagepar * 4)) ? ((h->lenpar) -
                                                                                          bar(230, 30, 450, 80);
(pagepar * 4)) : 4); i++)
                                                                                          puthz(240, 30, "最后一
                         {
                                                                 页啦", 32, 32, BLUE);
                              setfillstyle(1, 14);
                                                                                          delay(300);
                              bar(460, 150 + 60 * i,
                                                                                          setfillstyle(1, 0);
520, 150 + 40 + 60 * i);
                                                                                          bar(230, 30, 450, 80);
                              puthz(470, 160 + 60)
                                                                                          puthz(240, 30, "参数列
*i, "选择", 16, 16, BLUE);
                                                                 表", 32, 32, BLUE);
```

```
}
                                                                               int pagepar;
              }
                                                                               char page2[1];
                                                                               page2[1] = '\0';
             // 四个选择按钮
                                                                               page2[0] = page[0];
              if ((mouse press(460, 150 + 60 * 0,
                                                                               pagepar = atoi(page2) - 1;
520, 150 + 40 + 60 * 0) == 1) && ((h->lenpar -
                                                                               clrmous(MouseX, MouseY);
(pagepar * 4)) > 0))
              {
                                                                               return (pagepar)*4 + 4;
                  int pagepar;
                                                                          }
                  char page2[1];
                  page2[1] = '\0';
                                                                          // enter
                                                                          if (mouse press(0, 0, 40, 30) == 0
                  page2[0] = page[0];
                  pagepar = atoi(page2) - 1;
                                                            || mouse_press(0, 450, 40, 480) == 0 ||
                                                            mouse press(260, 395, 310, 415) == 0 ||
                                                            mouse_press(330, 395, 380, 415) == 0 ||
                  clrmous(MouseX, MouseY);
                  return (pagepar)*4 + 1;
                                                            mouse_press(510, 80, 580, 125) == 0)
              }
                                                                          {
              if ((mouse_press(460, 150 + 60 * 1,
                                                                               MouseS = 0;
520, 150 + 40 + 60 * 1) == 1) && ((h->lenpar -
                                                                          if (mouse_press(0, 0, 40, 30) == 2
(pagepar * 4)) > 1))
             {
                                                            || mouse press(0, 450, 40, 480) == 2 ||
                                                            mouse press(260, 395, 310, 415) == 2 ||
                  int pagepar;
                  char page2[1];
                                                            mouse_press(330, 395, 380, 415) == 2 ||
                  page2[1] = '\0';
                                                            mouse_press(510, 80, 580, 125) == 2)
                  page2[0] = page[0];
                                                                          {
                  pagepar = atoi(page2) - 1;
                                                                               MouseS = 1;
                  clrmous(MouseX, MouseY);
                                                                          }
                                                                          // quit
                  return (pagepar)*4 + 2;
                                                                          if (mouse\_press(0, 0, 40, 30) == 1)
              }
                                                                          {
              if ((mouse press(460, 150 + 60 * 2,
                                                                               exit(0);
520, 150 + 40 + 60 * 2 == 1) && ((h->lenpar -
                                                                          }
(pagepar * 4)) > 2))
                                                                          // last
                                                                          if (mouse_press(0, 450, 40, 480) ==
              {
                  int pagepar;
                                                            1)
                  char page2[1];
                                                                          {
                  page2[1] = '\0';
                                                                               return -1;
                  page2[0] = page[0];
                                                                          }
                  pagepar = atoi(page2) - 1;
                                                                     }
                  clrmous(MouseX, MouseY);
                                                                 }
                                                                 // 搜索
                  return (pagepar)*4 + 3;
                                                                 int search(char name[])
             if ((mouse_press(460, 150 + 60 * 3,
                                                                 {
520, 150 + 40 + 60 * 3) == 1) && ((h->lenpar -
                                                                      int i, j, flag = 0;
(pagepar * 4)) > 3))
                                                                      clrmous(MouseX, MouseY);
                                                                      cleardevice();
              {
```

```
clrmous(MouseX, MouseY);
                                                                                settextstyle(0, 0, 3);
          setbkcolor(WHITE);
                                                                                settextjustify(1, 1);
          quit();
                                                                                outtextxy(320,
                                                                                                     250,
                                                                                                                "the
                                                                parameter");
          setfillstyle(1, LIGHTGRAY);
                                                                                outtextxy(320, 200, "Can not find");
          bar(50, 80, 590, 420);
                                                                          }
          settextstyle(0, 0, 3);
          settextjustify(1, 1);
                                                                           setfillstyle(1, CYAN);
                                                                           bar(80, 120 + 60 * 4, 150, 160 + 60 * 4);
          outtextxy(320, 60, name);
                                                                           puthz(83, 125 + 60 * 4, "返回", 32, 32,
          for (i = 0; i < h->lenpar; i++)
                                                                BLUE);
                                                                          for (;;)
               for (j = 0; j < 10; j++) // 查找账号
位置
                                                                          {
               {
                                                                                newmouse(&MouseX,
                                                                                                         &MouseY,
                    if
                              (name[j]
                                                 !=
                                                                &press);
h->parameter[i].name[j])
                                                                                if (mouse_press(460, 150, 520, 150
                                                                + 40) == 1)
                   {
                                                                                {
                         break;
                   }
                                                                                     return i + 1;
                    if (name[j] == '\0')
                                                                                }
                                                                                // enter
                   {
                                                                                if (mouse\_press(0, 0, 40, 30) == 0
                        j = 10;
                                                                 || mouse_press(80, 120 + 60 * 4, 150, 160 + 60 *
                         break;
                                                                4) == 0
                   }
               }
                                                                                {
               if (j == 10)
                                                                                     MouseS = 0;
                                                                                if (mouse\_press(0, 0, 40, 30) == 2
                   flag = 1;
                    break;
                                                                 || mouse_press(80, 120 + 60 * 4, 150, 160 + 60 *
               }
                                                                4) == 2)
          }
                                                                                {
          if (flag == 1)
                                                                                     MouseS = 1;
          {
                                                                                }
                                                                                // quit
               setfillstyle(1, BLUE);
               bar(90, 150, 300, 150 + 40);
                                                                                if (mouse\_press(0, 0, 40, 30) == 1)
               setfillstyle(1, YELLOW);
               bar(460, 150, 520, 150 + 40);
                                                                                     exit(0);
               puthz(470, 160, "查看", 16, 16,
                                                                                }
BLUE);
                                                                                if (mouse press(80, 120 + 60 * 4,
               settextstyle(0, 0, 3);
                                                                150, 160 + 60 * 4) == 1)
               settextjustify(1, 1);
               outtextxy(195,
                                              170,
                                                                                {
h->parameter[i].name);
                                                                                     return -1;
         }
                                                                                }
          else
                                                                          }
          {
                                                                      }
```

```
setfillstyle(1, BROWN);
    void changewarename(int wi)
                                                                       bar(100, 130, 540, 370);
    {
                                                                       puthz(180, 150, "请重新输入参数名字",
         int i, j;
                                                             32, 32, BLUE);
         INPUT name = {185, 220, 455, 270, "", 8,
                                                                       setfillstyle(1, CYAN);
0, 0};
                                                                       bar(150, 290, 240, 340);
         clrmous(MouseX, MouseY);
                                                                       puthz(153, 293, "返回", 32, 32, BLUE);
         setfillstyle(1, CYAN);
                                                                       bar(400, 290, 490, 340);
         bar(100, 130, 540, 370);
                                                                       puthz(403, 293, "确认", 32, 32, BLUE);
         puthz(180, 150, "请重新输入仓库名字",
                                                                       setfillstyle(1, 0);
32, 32, BLUE);
                                                                       bar(175, 220, 465, 270);
         setfillstyle(1, GREEN);
         bar(150, 290, 240, 340);
                                                                       for (;;)
         puthz(153, 293, "返回", 32, 32, BLUE);
                                                                       {
         bar(400, 290, 490, 340);
                                                                            delay(15);
         puthz(403, 293, "确认", 32, 32, BLUE);
                                                                            newmouse(&MouseX,
                                                                                                    &MouseY,
         setfillstyle(1, 0);
                                                             &press);
         bar(185, 220, 455, 270);
                                                                            input_s(178, 220, &name, 16, 0);
         for (;;)
                                                                            if (mouse_press(150, 290, 240, 340)
         {
                                                             == 1) // 返回
              delay(15);
                                                                            {
              newmouse(&MouseX,
                                       &MouseY,
                                                                                 return;
&press);
              input s(188, 220, &name, 16, 0);
                                                                            if (mouse press(400, 290, 490, 340)
              if (mouse press(150, 290, 240, 340)
                                                             == 1) // 确认
== 1) // 返回
                                                                            {
                                                                                 strcpy(h->parameter[par
                                                             1].name, name.string);
                   return;
                                                                                 wr_h();
              if (mouse press(400, 290, 490, 340)
                                                                                 return;
== 1) // 确认
                                                                            }
                                                                       }
              {
                   strcpy(h->here[wi
                                                             }
1].ware name, name.string);
                                                     9、
                                                             PAST.C
                  wr_h();
                   return;
                                                                  #include "COMMON.H"
                                                                  #include "PAST.H"
              }
         }
                                                                  #include "PARAMETE.H"
    }
                                                                  #include "LOGFUN.H"
                                                                  #include "time.h"
    void changeparname(int par)
                                                                  void past01 screen(void)
    {
                                                                  {
                                                                   int i;
         int i, j;
         INPUT name = {175, 220, 465, 270, "",
                                                                    clrmous(MouseX, MouseY);
10, 0, 0};
                                                                    cleardevice();
         clrmous(MouseX, MouseY);
                                                                    clrmous(MouseX, MouseY);
```

```
setbkcolor(WHITE);
                                                                              page2[1] = '\0';
      puthz(240, 30, "参数列表", 32, 32, BLUE);
                                                                             page2[0] = page[0];
      quit();
                                                                             pagepar = atoi(page2) - 1;
      last();
                                                                             newmouse(&MouseX,
                                                                                                            &MouseY,
                                                                  &press);
      setfillstyle(1, LIGHTGRAY);
                                                                             if (h->lenpar != 0)
      bar(50, 80, 590, 420);
                                                                             {
                                                                                   input_s(223, 80, &searchname,
      settextstyle(0, 0, 2);
      settextjustify(1, 1);
                                                                  16, 0);
                                                                             }
                                                                             // flag 为 1 则换页
      setfillstyle(1, YELLOW);
                                                                             if (flag == 1)
      bar(505, 77, 580, 125);
    }
                                                                             {
                                                                                   if (h->lenpar == 0)
     int past01()
                                                                                   {
                                                                                        setfillstyle(1, LIGHTGRAY);
    {
      int i, j, flag = 1;
                                                                                        bar(50, 80, 590, 420);
              // 判断是否换页
                                                                                        settextjustify(1, 1);
      INPUT searchname = {220, 80, 500, 125, "",
                                                                                        settextstyle(0, 0, 4);
10, 0, 0}; // 搜索
                                                                                        setcolor(RED);
      char page[3] = {'1', '/', '1'};
                                                                                        outtextxy(320,
                                                                                                          250,
                                                                                                                  "No
      int barcolor[11] = {1, 2, 3, 4, 5, 9, 10, 11, 12,
                                                                  Parameter");
13, 14};
                                                                                        flag = 0;
                                                                                   }
      page[3] = '\0';
      for (i = 0; i < PAR; i++)
                                                                                   else
      {
                                                                                   {
           char a[1];
                                                                                        setfillstyle(1, LIGHTGRAY);
           a[1] = '\0';
                                                                                        bar(50, 80, 590, 420);
           if ((h->lenpar/4) == i)
                                                                                        setfillstyle(1, YELLOW);
           {
                                                                                        bar(505, 77, 580, 125);
                itoa(i, a, 10);
                                                                                       puthz(130, 83, "搜索", 32, 32,
                page[2] = a[0];
                                                                  BLUE);
                if ((h->lenpar % 4) != 0)
                                                                                       puthz(505, 83, "确认", 32, 32,
                                                                  RED);
                {
                     itoa(i + 1, a, 10);
                                                                                        settextjustify(1, 1);
                     page[2] = a[0];
                                                                                        setfillstyle(1, 0);
                }
                                                                                        bar(220, 80, 500, 125);
           }
                                                                                        rectangle(220, 80, 500, 125);
      }
                                                                                        settextstyle(0, 0, 2);
      past01_screen();
                                                                                        outtextxy(320, 405, page);
      delay(100);
                                                                                        outtextxy(280, 405, "<<");
      clrmous(MouseX, MouseY);
                                                                                        outtextxy(360, 405, ">>");
                                                                                        // settextstyle(0, 0, 3);
      for (;;)
                                                                                        setcolor(BLACK);
      {
                                                                   srand((unsigned)time(NULL));
           int pagepar;
                                                                                        for (i = 0, j = pagepar * 4; i <
           char page2[1];
```

```
for (;;)
(4 > ((h->lenpar) - (pagepar * 4)) ? ((h->lenpar) -
(pagepar * 4)) : 4); i++, j++)
                                                                                  {
                     {
                                                                                       int act;
                          int a, b;
                                                                                       act
                          a = barcolor[rand() %
                                                                 search(searchname.string);
                                                                                       if (act != -1)
12];
                          b = barcolor(rand() %
                                                                                       {
12];
                                                                                       past02there:
                          if (a == b)
                                                                                            if (past02(act) == -1)
                                                                                            {
                          {
                               a += 1;
                                                                                                 goto past02there;
                                                                                            }
                          setcolor(a);
                                                                                       }
                          setfillstyle(1, b);
                                                                                       else
                          bar(70, 150 + 60 * i, 320,
150 + 40 + 60 * i;
                                                                                            break;
                                                                                       }
(strlen(h->parameter[j].name) >= 7)
                                                                                  }
                                                                                  return -2;
                               settextstyle(0,
                                                 0,
                                                                            }
2);
                                                                            // page last
                          }
                                                                             if (mouse_press(260, 395, 310, 415) ==
                          else
                                                                 1)
                          {
                                                                            {
                               settextstyle(0,
                                                                                  if (page[0] > '1')
                                                 0,
3);
                                                                                  {
                          }
                                                                                       page[0] -= 1;
                          outtextxy(195, 170 + 60)
                                                                                       flag = 1;
* i, h->parameter[j].name);
                                                                                       delay(100);
                     }
                                                                                  }
                     for (i = 0; i < (4 > ((h->lenpar)))
                                                                                  else
- (pagepar * 4)) ? ((h->lenpar) - (pagepar * 4)) : 4);
                                                                                  {
i++)
                                                                                       setfillstyle(1, CYAN);
                     {
                                                                                       bar(230, 30, 450, 80);
                                                                                       puthz(240, 30, "第一页啦",
                          setfillstyle(1, 14);
                          bar(460, 150 + 60 * i,
                                                                 32, 32, BLUE);
520, 150 + 40 + 60 * i);
                                                                                       delay(300);
                          puthz(474, 160 + 60 * i,
                                                                                       setfillstyle(1, 0);
"查看", 16, 16, BLUE);
                                                                                       bar(230, 30, 450, 80);
                                                                                       puthz(240, 30, "参数列表",
                     }
                     flag = 0;
                                                                 32, 32, BLUE);
                }
                                                                                       rectangle(220, 80, 500, 125);
           }
                                                                                  }
           // search 返回-2
                                                                            }
           if (mouse_press(505, 77, 580, 125) ==
1)
                                                                            // page next
           {
                                                                             if (mouse press(330, 395, 380, 415) ==
```

```
1)
          {
                                                                       if ((mouse_press(460, 150 + 60 * 2,
               if (page[0] < page[2])
                                                            520, 150 + 40 + 60 * 2) == 1) && ((h->lenpar -
                                                            (pagepar * 4)) > 2))
               {
                   page[0] += 1;
                                                                      {
                   flag = 1;
                                                                           int pagepar;
                   delay(100);
                                                                           char page2[1];
                                                                           page2[1] = '\0';
               else if (page[0] == page[2])
                                                                           page2[0] = page[0];
                                                                           pagepar = atoi(page2) - 1;
                   setfillstyle(1, CYAN);
                   bar(230, 30, 450, 80);
                                                                           return (pagepar)*4 + 3;
                   puthz(240, 30, "最后一页啦",
                                                                      }
32, 32, BLUE);
                                                                       if ((mouse press(460, 150 + 60 * 3,
                   delay(300);
                                                            520, 150 + 40 + 60 * 3) == 1) && ((h->lenpar -
                   setfillstyle(1, 0);
                                                            (pagepar * 4)) > 3))
                   bar(230, 30, 450, 80);
                                                                      {
                   puthz(240, 30, "参数列表",
                                                                           int pagepar;
32, 32, BLUE);
                                                                           char page2[1];
                   rectangle(220, 80, 500, 125);
                                                                           page2[1] = '\0';
               }
                                                                           page2[0] = page[0];
          }
                                                                           pagepar = atoi(page2) - 1;
          // 四个查看按钮
                                                                           return (pagepar)*4 + 4;
          if ((mouse press(460, 150 + 60 * 0,
                                                                      }
520, 150 + 40 + 60 * 0) == 1) && ((h->lenpar -
(pagepar * 4)) > 0))
                                                                      // enter
                                                                       if (mouse_press(0, 0, 40, 30) == 0 ||
          {
                                                            mouse_press(0, 450, 40, 480) == 0 ||
               int pagepar;
               char page2[1];
                                                            mouse_press(260, 395, 310, 415) == 0 ||
                                                            mouse press(330, 395, 380, 415) == 0 ||
               page2[1] = '\0';
               page2[0] = page[0];
                                                            mouse press(510, 80, 580, 125) == 0
               pagepar = atoi(page2) - 1;
                                                                      {
                                                                           MouseS = 0;
               return (pagepar)*4 + 1;
          }
                                                                       if (mouse_press(0, 0, 40, 30) == 2 ||
                                                            mouse_press(0, 450, 40, 480) ==
          if ((mouse press(460, 150 + 60 * 1,
520, 150 + 40 + 60 * 1) == 1) && ((h->lenpar -
                                                            mouse press(260, 395, 310, 415) == 2 ||
(pagepar * 4)) > 1))
                                                            mouse_press(330, 395, 380, 415) == 2 ||
                                                            mouse_press(510, 80, 580, 125) == 2)
          {
               int pagepar;
                                                                      {
               char page2[1];
                                                                           MouseS = 1;
               page2[1] = '\0';
                                                                      }
                                                                       // quit
               page2[0] = page[0];
               pagepar = atoi(page2) - 1;
                                                                      if (mouse\_press(0, 0, 40, 30) == 1)
                                                                      {
               return (pagepar)*4 + 2;
                                                                           exit(0);
```

```
}
                                                                         puthz(240, 120 + 60, "圆形", 32, 32,
           // last
                                                              BLUE);
           if (mouse press(0, 450, 40, 480) == 1)
                                                                    else if (h->parameter[par - 1].shape == 'c')
           {
                return -1;
                                                                         puthz(240, 120 + 60, "多边形", 32, 32,
          }
      }
                                                              BLUE);
    }
                                                                     else if (h->parameter[par - 1].shape == 'd')
    void past02 screen(int par)
                                                                         puthz(240, 120 + 60, "自定义图形", 32,
                                                              32, BLUE);
      int i;
      clrmous(MouseX, MouseY);
                                                                    }
      cleardevice();
      setbkcolor(WHITE);
                                                                     puthz(80, 120 + 60 * 2, "收割机类型: ", 32,
      settextstyle(0, 0, 3);
                                                              32, BLUE);
      settextjustify(1, 1);
                                                                    if (h->parameter[par - 1].type == 'a')
      setcolor(BLUE);
      outtextxy(320, 45, h->parameter[par -
                                                                         puthz(260, 120 + 60 * 2, "垂直式收割
1].name);
                                                              机", 32, 32, BLUE);
      quit();
                                                                    else if (h->parameter[par - 1].type == 'b')
      setfillstyle(1, LIGHTGRAY);
                                                                         puthz(260, 120 + 60 * 2, "水平式收割
      bar(50, 100, 590, 420);
      puthz(80, 120, "产区: ", 32, 32, BLUE);
                                                              机", 32, 32, BLUE);
      if (h->parameter[par - 1].place == 'a')
                                                                    }
      {
           puthz(180, 120, "新疆", 32, 32, BLUE);
                                                                     puthz(80, 120 + 60 * 3, "面积: ", 32, 32,
                                                              BLUE):
      }
      else if (h->parameter[par - 1].place == 'b')
                                                                     settextjustify(0, 2);
                                                                    outtextxy(180,
                                                                                      130
                                                                                                   60
                                                                                                              3,
      {
           puthz(180, 120, "黄河", 32, 32, BLUE);
                                                              h->parameter[par - 1].S);
      }
                                                                     puthz(320, 120 + 60 * 3, "公顷", 32, 32,
      else if (h->parameter[par - 1].place == 'c')
                                                              BLUE);
           puthz(180, 120, "长江", 32, 32, BLUE);
                                                                     setfillstyle(1, YELLOW);
      }
                                                                    for (i = 0; i < 4; i++)
      puthz(80, 120 + 60, "土地形状: ", 32, 32,
                                                                         bar(500, 118 + 60 * i, 550, 118 + 40 +
BLUE);
                                                              60 * i);
      if (h->parameter[par - 1].shape == 'a')
                                                                         puthz(510, 130 + 60 * i, "修改", 16, 16,
                                                              BLUE);
           puthz(240, 120 + 60, "矩形", 32, 32,
BLUE);
                                                                    setfillstyle(1, CYAN);
                                                                     bar(80, 120 + 60 * 4, 150, 160 + 60 * 4);
      }
                                                                     puthz(83, 125 + 60 * 4, "返回", 32, 32,
      else if (h->parameter[par - 1].shape == 'b')
      {
                                                              BLUE);
```

```
return -1;
      bar(400, 120 + 60 * 4, 550, 160 + 60 * 4);
                                                                          }
     puthz(408, 125 + 60 * 4, "修改名字", 32, 32,
                                                                          if (mouse press(500, 118 + 60 * 2, 550,
BLUE);
                                                               118 + 40 + 60 * 2) == 1) // type
      setfillstyle(1, RED);
                                                                               changetype(par);
      bar(400 + 40, 30, 480 + 40, 80);
                                                                               return -1;
      setcolor(BLACK);
      rectangle(399 + 40, 29, 481 + 40, 81);
                                                                          if (mouse_press(500, 118 + 60 * 3, 550,
      puthz(409 + 40, 38, "删除", 32, 32, BLUE);
                                                               118 + 40 + 60 * 3) == 1) // S
    }
                                                                          {
                                                                               changeS(par);
    int past02(int par)
                                                                               return -1;
                                                                          }
      int i;
      past02_screen(par);
                                                                          // quit,last
                                                                          if (mouse_press(0, 0, 40, 30) == 0 ||
      for (;;)
                                                               mouse_press(80, 120 + 60 * 4, 150, 160 + 60 * 4)
      {
                                                               == 0)
           newmouse(&MouseX,
                                        &MouseY,
                                                                          {
&press);
                                                                               MouseS = 0;
                                                                          }
           // 删除按钮
                                                                          if (mouse_press(0, 0, 40, 30) == 2 ||
          if (mouse_press(400 + 40, 30, 480 + 40,
                                                               mouse_press(80, 120 + 60 * 4, 150, 160 + 60 * 4)
80) == 1)
                                                               == 2)
           {
                                                                          {
                deletepar(par);
                                                                               MouseS = 1;
                return 1;
           }
                                                                          if (mouse\_press(0, 0, 40, 30) == 1)
           // 改名字
                                                                          {
          if (mouse press(400, 120 + 60 * 4, 550,
                                                                               exit(0);
160 + 60 * 4) == 1)
                                                                          }
           {
                changeparname(par);
                                                                          if (mouse press(80, 120 + 60 * 4, 150,
                return -1;
                                                               160 + 60 * 4) == 1
           }
           // 四个修改按钮
                                                                          {
          if (mouse press(500, 118 + 60 * 0, 550,
                                                                               return 1;
118 + 40 + 60 * 0) == 1) // place
           {
                                                                          delay(15);
                changeplace(par);
                                                                     }
                return -1;
                                                                    }
                                                                    void past()
          if (mouse_press(500, 118 + 60 * 1, 550,
118 + 40 + 60 * 1) == 1) // shape
                                                                     int act, i;
           {
                changeshape(par);
                                                                     for (;;)
```

```
{
                                                                      puthz(160, 423, "返回", 32, 32, RED);
          act = past01();
                                                                      bar(360, 420, 520, 460); // 确认
          if (act == -1) // 返回主界面
                                                                      puthz(400, 423, "确认", 32, 32, RED);
          {
               return;
                                                                      setfillstyle(1, WHITE);
                                                                     bar(260, 340, 530, 390);
          }
          else if (act == -2) // 搜索函数
                                                                     bar(260, 260, 530, 310);
                                                                     bar(220, 180, 530, 230);
               continue;
                                                                     bar(220, 100, 530, 150);
                                                                 }
      there:
          if (past02(act) == -1)
                                                                 void log register()
          {
                                                                 {
               goto there;
                                                                     // 界面
          }
                                                                     INPUT username = {220, 100, 530, 150,
     }
                                                            "", 10, 0, 1};
}
                                                                     INPUT password = {220, 180, 530, 230,
                                                            "", 10, 0, 1};
REGISTER.C
                                                                      INPUT realpassword = {260, 260, 530,
    #include "COMMON.H"
                                                            310, "", 10, 0, 1};
    #include "REGISTER.H"
                                                                     INPUT phonenumber = {260, 340, 530,
    #include "PARAMETE.H"
                                                            390, "", 11, 0, 1};
    #include "LOGFUN.H"
                                                                     register_screen();
    void register_screen()
                                                                     for (;;)
    {
         cleardevice();
                                                                     {
         setbkcolor(WHITE);
                                                                          newmouse(&MouseX,
                                                                                                  &MouseY,
         setfillstyle(1, LIGHTBLUE);
                                                            &press);
         bar(80, 70, 560, 500);
                                                                          input_s(260, 260, &realpassword,
                                                            16, 1);
         setfillstyle(1, LIGHTGRAY);
                                                                          if (input_s(223, 100, &username,
         puthz(260, 30, "注册账号", 32, 32,
                                                            16, 0) == 1)
BLUE);
                                                                          {
         bar(120, 105, 530, 150); // 账号
                                                                               setfillstyle(1, CYAN);
         puthz(123, 110, "账号", 32, 32, BLUE);
                                                                               bar(170, 30, 470, 72);
                                                                               puthz(180, 32, "账号名最多
         bar(120, 180, 530, 230); // 密码
                                                            十位字符", 32, 32, BLUE);
         puthz(123, 190, "密码", 32, 32, BLUE);
         bar(120, 260, 530, 310); // 确认密码
                                                                               delay(800);
         puthz(123, 270, "确认密码", 32, 32,
                                                                               setfillstyle(1, 0);
BLUE);
                                                                               bar(170, 30, 470, 72);
         bar(120, 340, 530, 390); // 电话号码
                                                                               puthz(260, 30, "注册账号", 32,
         puthz(123, 350, "电话号码", 32, 32,
                                                            32, BLUE);
BLUE);
                                                                          if (input_s(220, 180, &password, 16,
         setfillstyle(1, DARKGRAY);
                                                            1) == 1)
                                                                          {
         bar(120, 420, 280, 460); // 返回
```

10

```
}
                  setfillstyle(1, CYAN);
                  bar(170, 30, 470, 72);
                                                    11、
                                                            RESET.C
                  puthz(180, 32, "密码最多十
位字符", 32, 32, BLUE);
                                                                 #include "COMMON.H"
                                                                 #include "LOGIN.H"
                  delay(800);
                                                                 #include "PARAMETE.H"
                  setfillstyle(1, 0);
                  bar(170, 30, 470, 72);
                                                                 #include "RESET.H"
                  puthz(260, 30, "注册账号", 32,
                                                                 #include "LOGFUN.H"
32, BLUE);
             }
             if
                                           340,
                                                                 void reset_screen()
                      ((input_s(263,
&phonenumber, 16, 0))==1)
                                                                     cleardevice();
             {
                  setfillstyle(1, CYAN);
                                                                     setbkcolor(WHITE);
                  bar(170, 30, 470, 72);
                                                                     setfillstyle(1,LIGHTBLUE);
                  puthz(180, 32, "请输入十一
                                                                     bar(80,80,560,500);
位号码", 32, 32, BLUE);
                                                                     setfillstyle(1,LIGHTGRAY);
                  delay(800);
                                                                     puthz(260,30,"忘记密码",32,32,BLUE);
                  setfillstyle(1, 0);
                  bar(170, 30, 470, 72);
                                                                     bar(120,
                                                                                   140,
                                                                                             520,
                                                                                                       200);
                  puthz(260, 30, "注册账号", 32,
                                                            //账号
32, BLUE);
                                                                     puthz(123,150,"账号",32,32,BLUE);
             }
                                                                     bar(120,
                                                                                   220,
                                                                                             520,
                                                                                                       280);
             if (mouse_press(120, 420, 280, 460)
                                                            //电话号码
== 1)
                                                                     puthz(123,230,"电话号码",32,32,BLUE);
             {
                                                                     bar(120,
                                                                                   300,
                                                                                             520,
                                                                                                       360);
                  return;
                                                            //新密码
                                                                     puthz(123,310,"新密码",32,32,BLUE);
              if ((mouse_press(360, 420, 520,
460) == 1) &&
                                                                     setfillstyle(1,DARKGRAY);
                  (username.string[0] != '\0')
                                                                                                       460);
                                                                     bar(120,
                                                                                   420,
                                                                                             280,
                                                            //返回
&& (phonenumber.string[0] != '\0') &&
                  (password.string[0] != '\0') \&\&
                                                                     puthz(160,423,"返回",32,32,BLUE);
(strcmp(password.string, realpassword.string) ==
                                                                     bar(360,
                                                                                   420,
                                                                                             520,
                                                                                                       460);
                                                            //确认
0))
             {
                                                                     puthz(400,423,"确认",32,32,BLUE);
(username same(username.string,
                                                                     setfillstyle(1,WHITE);
phonenumber.string) == 0)
                                                                     bar(220, 140, 520, 200);
                  {
                                                                     bar(260, 220, 520, 280);
                       wr user(username.string,
                                                                     bar(220, 300, 520, 360);
password.string, phonenumber.string);
                                                                 }
                       return;
                  }
             }
                                                                 void reset()
              delay(15);
                                                                 {
         }
                                                                     INPUT username = {220, 140, 520,
```

```
200,"",10,0,1};
                                                                  //{
       INPUT phonenumber = {260, 220, 520,
                                                                  //
                                                                        int gd=VGA,gm=VGAHI;
 280,"",11,0,1};
                                                                  //
          INPUT newpassword = {220, 300, 520,
                                                               initgraph(&gd,&gm,"..\borlandc\\bgi");
 360,"",10,0,1};
                                                                        setbkcolor(WHITE);
                                                                  //
                                                                  //
                                                                        h=(user*)malloc(sizeof(user));// 登 录
                                                              的用户
          reset screen();
                                                                  //
                                                                        //select03(&(h->parameter[1]));
          for(;;)
                                                                  //
                                                                        strcpy(h->parameter[0].name,"abc");
          {
                                                                  //
                                                                        strcpy(h->parameter[1].name,"acc");
                                                                  //
                                                                        //strcpy(&h->parameter[0].type,"a");
 newmouse(&MouseX,&MouseY,&press);
                                                                        h->parameter[1].type='a';
                                                                  //
               input_s(223, 140, &username, 12,
                                                                  //
                                                                        h->parameter[1].place='a';
 0);
                                                                  //
                                                                        //strcpy(&h->parameter[0].shape,"a");
               input s(263, 220, &phonenumber,
                                                                  //
                                                                        h->parameter[1].shape='a';
 12,1);
                                                                        strcpy(h->parameter[1].S,"500");
                                                                  //
               input_s(223, 300, &newpassword,
                                                                  //
                                                                        start();
 12,1);
                                                                        delay(5000);
                                                                  //
               if(mouse press(120,
                                     420.
                                             280,
                                                                  //
                                                                        return 0;
 460)==1)
                                                                  // }
               {
                                                                  // int main()
                                                                  //{
                    return;
                                                                  //
                                                                        int gd=VGA,gm=VGAHI;
               }
                                                                  //
               if(mouse_press(360,
                                     420,
                                            520,
 460) = = 1)
                                                              gd=VGA,gm=VGAHI,start_x=400,start_y=100,des_
               {
                                                              x=100, des y=300;
                                                                  //
 if(changepassword(username.string,
                                                               initgraph(&gd,&gm,"..\\borlandc\\bgi");
 newpassword.string, phonenumber.string)==1)
                                                                  //
                                                                        setbkcolor(WHITE);
                   {
                                                                  //
                                                                        h=(user*)malloc(sizeof(user));// 登 录
                                                              的用户
                        return;
                   }
                                                                  //
                                                                        strcpy(h->parameter[0].name,"abc");
               }
                                                                  //
                                                                        strcpy(h->parameter[1].name,"acc");
               delay(15);
                                                                        //strcpy(&h->parameter[0].type,"a");
                                                                  //
          }
                                                                  //
                                                                        h->parameter[1].type='a';
 }
                                                                  //
                                                                        h->parameter[1].place='a';
                                                                  //
                                                                        //strcpy(&h->parameter[0].shape,"a");
START.C
                                                                  //
                                                                        h->parameter[1].shape='b';
      #include "START.H"
                                                                  //
                                                                        strcpy(h->parameter[1].S,"300");
      #include "COMMON.H"
                                                                  //
                                                                        start();
      #include "PARAMETE.H"
                                                                  //
                                                                        delay(5000);
      #include "PAST.H"
                                                                  //
                                                                        return 0;
      #include "DSTART.H"
                                                                  //}
      #include "LOGFUN.H"
                                                                  // int main()
      #include "HOME.H"
                                                                  //{
      int delaytime=20;
                                                                  //
                                                                        int gd=VGA,gm=VGAHI,x=300,y=200;
      // int main()
                                                                  //
```

12、

```
initgraph(&gd,&gm,"..\\borlandc\\bgi");
         setbkcolor(BROWN);
                                                              harvest=space*Huanghe_har*(1+(rand()%8)/100
   //
   //
         mouseinit();
                                                             );
         picker_anime(50,50,&x,&y,10,1);
   //
                                                                             co_type=1;
   //
         delay(5000);
                                                                             break;
   //
                                                                        }
         return 0;
                                                                        case 'c':
   //}
                                                                        {
   void start()
                                                              harvest=space*Chnagjiang har*(1+(rand()%8)/1
    int i=0,time=0,co_type=0;
                                                             00);
    double space=0,harvest=0;
                                                                             co_type=2;
    delaytime=20;
                                                                             break;
    // for(i=0;i<10;i++)
                                                                        }
    //{
                                                                   }
    //
                                                                   if(harvest==0)
if(strcmp(h->parameter[i].name,"\0")==0)
                                                                        harvest=3;
    //
        {
    //
                                                                   switch(h->parameter[i].type)
              break;
    // }
                                                                   {
    //}
                                                                        case 'a':
    // i--;
                                                                        {
    i=choosepar();
                                                                             //draw_simu01(time);
                                                                             switch(h->parameter[i].shape)
    i--;
     delay(500);
                                                                             {
     space=atoi(h->parameter[i].S);
                                                                                  case 'a':
     if(space>32767||space<=0)
                                                                                  {
         space=32767;
     if(space<500)
                                                              start_ainime01(0,space,time);
         space=500;
                                                                                       break;
    time=space/pick ph;
                                                                                  }
     if(time<tra_time)
                                                                                  case 'b':
         time=tra_time;
    if(time>tracktor_num_max*tra_time)
         time=tracktor num max*tra time;
                                                              cal tracktor circle(0,space,time);
    switch(h->parameter[i].place)
                                                                                       break;
    {
                                                                                  }
                                                                                  case 'c':
         case 'a':
         {
                                                                                  {
                                                              init_field02(h->parameter[i].x,h->parameter[i].y,
harvest=space*Xinjinag_har*(1+(rand()%8)/100)
                                                             &(h->parameter[i].lenxy),0,time);
                                                                                       break;
              co_type=0;
              break;
                                                                                  }
         }
                                                                                  case 'd':
         case 'b':
         {
```

```
init_field03(h->parameter[i].x,h->parameter[i].y,
&(h->parameter[i].lenxy),0,time);
                                                                            default:
                          break;
                                                                                break;
                     }
                                                                      }
                }
                if(mode!=3)
                                                                      // switch (h->parameter[i].place)
                     return;
                pick_finish(harvest,co_type);
                                                                      //{
                break;
                                                                      //
                                                                           case 'a':
                                                                      //
                                                                           {
           }
                                                                                h->here[k].total[0]+=harvest;
                                                                      //
           case 'b':
                                                                      //
                                                                                break;
           {
                                                                           }
                                                                      //
                switch(h->parameter[i].shape)
                                                                      //
                                                                           case 'b':
                                                                      //
                                                                           {
                     case 'a':
                                                                      //
                                                                                h->here[k].total[1]+=harvest;
                     {
                                                                                break;
                                                                      //
                                                                      //
                                                                           }
 start_ainime01(1,space,time*0.8);
                                                                      //
                                                                           case 'c':
                          break;
                                                                      //
                                                                           {
                     }
                                                                      //
                                                                                h->here[k].total[2]+=harvest;
                     case 'b':
                                                                      //
                                                                                break;
                     {
                                                                      //
                                                                           }
 cal_tracktor_circle(1,space,time*0.8);
                                                                      //
                                                                           default:
                          break;
                                                                      //
                                                                                break;
                     }
                                                                      //}
                     case 'c':
                                                                      return;
                     {
                                                                     }
 init_field02(h->parameter[i].x,h->parameter[i].y,
&h->parameter[i].lenxy,1,time*0.8);
                                                                     //draw the process of harvest in animition
                                                                     void draw_simu01(int time)
                          break;
                     }
                                                                     {
                     case 'd':
                                                                      char str[8];
                                                                      itoa(time,str,10);
                     {
                                                                       clrmous(MouseX,MouseY);
 init_field03(h->parameter[i].x,h->parameter[i].y,
                                                                          cleardevice();
&h->parameter[i].lenxy,1,time*0.8);
                                                                          setbkcolor(WHITE);
                          break;
                                                                       puthz(150,30,"采摘完成共需",32,32,BLUE);
                     }
                                                                       settextstyle(3,0,4);
                }
                                                                       setcolor(RED);
                if(mode!=3)
                                                                       settextjustify(0,2);
                     return;
                                                                       outtextxy(350,28,str);
                pick_finish(harvest*0.8,co_type);
                                                                          puthz(450,30,"小时",32,32,BLUE);
                break;
                                                                       init_based_field();
           }
                                                                          //outtextxy()
```

```
quit();
                                                               s_y[2*tracktor_num_max];
      skip();
                                                                     double temp_x,temp_y;
         // mouseinit();
                                                                     temp y=sqrt(space/55*32)*10;
      // start_ainime01(0,x,y,num);
                                                                     temp_x=temp_y*55/32;
      // pick finish(temp);
                                                                     x=temp_x,y=temp_y,num=time/tra_time;
      // tal[c_t]+=temp;
                                                                     if(x>x_max)
      // if(tal[c_t]>ware_full||tal[c_t]<0)</pre>
                                                                          x=x_max;
      //{
                                                                     if(y>y_max)
      // tal[c_t]=ware_full;
                                                                          y=y_max;
      //}
                                                                     if(num==0)
      // for(;;)
                                                                          num=1;
      //{
                                                                     for(i=0;i<num;i++)
      //
                                                                     {
 newmouse(&MouseX,&MouseY,&press);
                                                                          des x[i]=600;
          press_start();
                                                                          des_y[i]=0;
      //
          delay(20);
                                                                     }
      //}
                                                                     i=0,x_p=x_start,y_p=y_start+y-40,flag=0,ou
                                                               t=0;
      //bmp_convert(".\\photo\\map.bmp",".\\p
                                                                     select_setoff01(xy,x_start+x,y_start+y);
hoto\\map.dbm");
                                                                     draw simu01(time);
      //show dbm(5,100,".\\photo\\map.dbm");
                                                                     clrmous(MouseX,MouseY);
      //getchar();
                                                                     draw setoff(xy);
      //closegraph();
                                                                     init_field(x,y);
      return;
                                                                     x/=num;
                                                                     if(t_trac==0)
    }
                                                                     {
    void init_based_field()
                                                                          draw_setoff(xy);
      setfillstyle(1,BROWN);
                                                                tracktor_set_off(xy[0],xy[1],x_start,y_start+y,x,n
      bar(0,80,625,480);
                                                               um);
      return;
                                                                          earth_fill01(x_p, y_p);
    }
                                                                          init_tracktor01_f(x_p, y_p);
                                                                          while (1)
    void draw_copak(int x,int y)
                                                                          {
                                                                if(cal_time>0&&cal_time>=pick_bar)
      if(x<x_start||y<y_start)</pre>
           return;
                                                                               {
      setfillstyle(1,WHITE);
                                                                                    cal time=-1;
      bar(x,y,x+tracktor_w,y+co_pak_w);
                                                                                    for(i=0;i<num;i++)
    }
                                                                                   {
                                                                                         if(flag==0)
    //start playing the picking video
                                                                                         {
    void
             start ainime01(int
                                    t trac, double
                                                                draw_copak(x_p+i*x,y_p+tracktor_l+co_pak_w);
space, int time)
                                                                                             des_x[i]=x_p+i*x;
      int i,x_p,y_p,flag,out,x,y,num,xy[2],\
      cal_time=0,des_x[2*tracktor_num_max],de
                                                                des y[i]=y p+tracktor l+co pak w;
```

```
}
                           else
                                                                                              if (y_p \le 120 + y - 45)
                           {
                                                                                                    earth_fill02(x_p + i
 draw_copak(x_p+i*x,y_p-2*co_pak_w);
                                                                   * x, y_p);
                                des_x[i]=x_p+i*x;
                                                                    init_tracktor01_b(x_p + i * x, y_p);
 des_y[i]=y_p-2*co_pak_w;
                                                                                              if (y_p >= 120 + y - 45)
                      }
                                                                                              {
                                                                                                    earth_fill03(x_p + i
                 if(cal_time>=0)
                                                                   * x, 120 + y - 49);
                      cal_time++;
                                                                                              }
                 if (x_p >= 50 + x)
                                                                                         delay(delaytime / num);
                      break;
                                                                                    }
                                                                                    if (flag == 0)
                 for (i = 0; i < num; i++)
                                                                                    {
                                                                                         y_p--;
                                                                                         if (y_p \le 120 + 7)
                      newmouse(&MouseX,
&MouseY, &press);
                      if (flag == 0)
                                                                                              for (i = 0; i < num; i++)
                      {
                                                                                                    earth_fill03(x_p + i
                                                                   * x, 120);
                           if (y_p == 120 + y - 45)
                                                                                              flag = 1;
                                earth fill03(x p + i
                                                                                              x p += 25;
* x, y_p - 4);
                                                                                         }
                                                                                    }
                           if (y_p >= 120 + 7)
                                                                                    else
                           {
                                                                                    {
                                earth_fill01(x_p + i
                                                                                         y_p++;
* x, y_p);
                                                                                         if (y_p >= 120 + y - 45)
 init_tracktor01_f(x_p + i * x, y_p);
                                                                                              for (i = 0; i < num; i++)
                                                                                                    earth fill03(x p + i
                           if (y_p \le 120 + 7)
                                                                   * x, 120 + y - 49);
                                                                                              flag = 0;
                                earth_fill03(x_p + i
                                                                                              x_p += 25;
* x, 120);
                                                                                         }
                           }
                                                                                    }
                      }
                                                                                    out
                                                                   pressed_anime(x_start,y_start,x * num, y);
                      else
                      {
                                                                                    if (out != 0)
                           if (y_p == 120 + 7)
                                                                                    {
                                                                                         // for (i = 1; i <= num; i++)
                                earth_fill01(x_p + i
                                                                                         //{
                                                                                         //
* x, y_p - 7);
```

```
init_tracktor01_f(x_start + i * x - x % 25, y_start +
7);
                                                                  draw_copak(x_p+i*x,y+tracktor_l+45);
                     //}
                                                                                                des x[i]=x p+i*x;
                     break;
                                                                  des y[i]=y+tracktor l+45;
                for(i=0;i<num;i++)
                                                                                           }
                                                                                           else
 draw_copak(des_x[i],des_y[i]);
                                                                  draw copak(x p+i*x,y-45-co pak w);
                // delay(15*num);
                                                                                                des_x[i]=x_p+i*x;
           }
                                                                  des_y[i]=y-45-co_pak_w;
 tracktor_return(xy[0],xy[1],x_start+x,y_start,x,nu
m);
                                                                                      }
                                                                                 }
 picker_anime(xy[0],xy[1],des_x,des_y,x,num);
                                                                                 if(cal_time>=0)
           return;
                                                                                       cal_time++;
           // for(i=1;i<=num;i++)
                                                                                 if (x p >= 50 + x)
           //{
                                                                                 {
           // if (flag == 1)
                                                                                       break;
                     init_tracktor01_f(x_start + i *
           //
x - x % 25, y_start + 7);
                                                                                 for (i = 0; i < num; i++)
           // else
                                                                                 {
           //
                     init_tracktor01_b(x_start + i
                                                                                       newmouse(&MouseX,
* x - x % 25, y start + y - 45);
                                                                 &MouseY, &press);
           //}
                                                                                      if (flag == 0)
      }
                                                                                           if (y_p == 120 + y - 45)
      else
      {
                                                                                           {
           clrmous(MouseX,MouseY);
                                                                                                earth fill03(x p + i
           draw_setoff(xy);
                                                                 * x, y_p - 4);
                                                                                           }
 tracktor_set_off0(xy[0],xy[1],x_start,y_start+y,x,
                                                                                           if (y_p >= 120 + 7)
num);
           earth_fill01(x_p, y_p);
                                                                                                earth_fill01(x_p + i
           init_tracktor02_f(x_p, y_p);
                                                                 * x, y_p);
           while (1)
                                                                  init\_tracktor02\_f(x\_p + i * x, y\_p);
           {
                                                                                           }
 if(cal_time>0&&cal_time>=pick_bar)
                                                                                           if (y_p \le 120 + 7)
                     cal time=-1;
                                                                                                earth_fill03(x_p + i
                     for(i=0;i<num;i++)
                                                                 * x, 120);
                                                                                           }
                          if(flag==0)
                                                                                      }
                          {
                                                                                       else
```

```
{
                                                                                  if (out != 0)
                          if (y_p == 120 + 7)
                                                                                  {
                                                                                       // for (i = 1; i <= num; i++)
                          {
                               earth_fill01(x_p + i
                                                                                       //{
                                                                                       //
* x, y_p - 7);
                                                                   init_tracktor02_f(x_start + i * x - x % 25, y_start +
                          }
                          if (y_p \le 120 + y - 45)
                                                                 7);
                                                                                       //}
                               earth_fill02(x_p + i
                                                                                       break;
* x, y_p);
                                                                                  }
                                                                                  for(i=0;i<num;i++)
 init_tracktor02_b(x_p + i * x, y_p);
                          if (y_p >= 120 + y - 45)
                                                                   draw_copak(des_x[i],des_y[i]);
                                earth_fill03(x_p + i
                                                                                  // delay(15*num);
* x, 120 + y - 49);
                                                                             }
                          }
                     }
                                                                   tracktor_return0(xy[0],xy[1],x_start+x,y_start,x,n
                     delay(delaytime/num);
                                                                 um);
                if (flag == 0)
                                                                   picker_anime(xy[0],xy[1],des_x,des_y,x,num);
                                                                             return;
                                                                             // for(i=1;i<=num;i++)
                     y_p--;
                     if (y_p \le 120 + 7)
                                                                             //{
                                                                             // if (flag == 1)
                     {
                          for (i = 0; i < num; i++)
                                                                             //
                                                                                       init_tracktor02_f(x_start + i *
                               earth_fill03(x_p + i
                                                                 x - x % 25, y_start + 7);
* x, 120);
                                                                             // else
                          flag = 1;
                                                                             //
                                                                                       init_tracktor02_b(x_start + i
                          x p += 25;
                                                                 * x - x % 25, y_start + y - 45);
                     }
                                                                             //}
                }
                                                                        }
                else
                                                                      }
                {
                                                                      //add the press moudules in start page
                     y_p++;
                                                                      void press_start(int *bk)
                     if (y_p >= 120 + y - 45)
                          for (i = 0; i < num; i++)
                                                                        if (mouse_press(0, 0, 40, 30) == 0 ||
                               earth_fill03(x_p + i
                                                                 mouse_press(265, 350, 365, 410) == 0)
* x, 120 + y - 49);
                                                                        {
                          flag = 0;
                                                                             MouseS = 0;
                          x p += 25;
                     }
                                                                        if (mouse_press(0, 0, 40, 30) == 2 ||
                                                                 mouse_press(265, 350, 365, 410) == 2)
                }
                                                                        {
                out
pressed_anime(x_start,y_start,x * num, y);
                                                                             MouseS = 1;
```

```
}
                                                                        *times+=0.1;
                                                                   //
      if (mouse_press(0, 0, 40, 30) == 1)
                                                                   //}
                                                                   if (mouse\_press(0, 0, 40, 30) == 1)
          wr_h();
                                                                   {
          free(h);
                                                                        re=1;
          exit(0);
                                                                        wr_h();
      }
                                                                        free(h);
      else if (mouse_press(265, 350, 365, 410) ==
                                                                        exit(0);
1)
                                                                   }
      {
                                                                   if (mouse press(585, 450, 625, 480) == 1)
          //draw_home01();
          mode=1;
                                                                        anime_skip_result(x_sta,
          mode1=0;
                                                             y_sta,x_des,y_des);
          *bk=1;
                                                                        re = 1;
          return;
                                                                   }
      }
                                                                   return re;
    }
                                                                  }
    // add press moudules in video page
                                                                  // show after picking
    int pressed_anime(int x_sta,int y_sta, int
                                                                  void pick_finish(int count,int co_type)
x des,int y des)
                                                                  {
                                                                   char str[10];
    {
      int re;
                                                                   int bk=0,re=0;
                                                                   itoa(count,str,10);
      re=0;
      if(MouseX>=x_sta&&MouseY>=y_sta&&Mo
                                                                   setfillstyle(1,WHITE);
useX<=x sta+x des&&MouseY<=y sta+y des)
                                                                   bar(160,170,470,340);
      {
                                                                   setcolor(RED);
          clrmous(MouseX, MouseY);
                                                                   setlinestyle(0, 0, 3);
      }
                                                                   rectangle(165, 175, 465, 335);
      if (mouse_press(0, 0, 40, 30) == 0 ||
mouse_press(585, 450, 625, 480) == 0)
                                                                   puthz(210,190,"采摘完成",32,32,BLUE);
      {
                                                                   puthz(210,230,"共计: ",32,32,BLUE);
          MouseS = 0;
                                                                   settextstyle(3,0,4);
      }
                                                                   puthz(380,230,"吨",32,32,BLUE);
      if (mouse press(0, 0, 40, 30) == 2 | |
                                                                   settextjustify(0,2);
mouse_press(585, 450, 625, 480) == 2)
                                                                   outtextxy(300,225,str);
      {
                                                                   switch(co_type)
          MouseS = 1;
                                                                   {
      }
                                                                   case 0:
      //
                                if(*times>0.5&&
                                                                   {
bioskey(0)==p_Up_arrow)
                                                                       puthz(210, 270, "种类:长绒棉", 32, 32,
      //{
                                                             BLUE);
      // *times-=0.1;
                                                                        break;
      //}
                                                                   }
      //
                                                                   case 1:
if(*times<2&&bioskey(0)==p_Donw_arrow)
      //{
                                                                       puthz(210, 270, "种类:细绒棉", 32, 32,
```

```
BLUE);
                                                                 the result
           break;
                                                                      void anime_skip_result(int x_sta,int y_sta, int
      }
                                                                 x_des,int y_des)
      case 2:
                                                                      {
                                                                        //int i = 0;
           puthz(210, 270, "种类:粗绒棉", 32, 32,
                                                                        setfillstyle(1, BROWN);
BLUE);
                                                                        setcolor(WHITE);
           break;
                                                                        bar(x_sta, y_sta, x_des, y_des);
      }
                                                                        // for (i = 0; i < x*y*0.05; i++)
      default:
                                                                       //{
           break;
                                                                       // int x_t = rand() \% x, y_t = rand() \% y;
      }
                                                                       // line(x_start + x_t, y_start + y_t, x_start
                                                                 + x_t, y_start + y_t);
      setfillstyle(1,YELLOW);
                                                                       //}
      bar(265,350,365,410);
                                                                      }
      puthz(278,360,"入库",32,32,RED);
      while(1)
                                                                      // initialize the cotton field
                                                                      void init_field(int x, int y)
      {
           if(bk!=0)
                break;
                                                                        int i,j;
                                                                        setfillstyle(1,WHITE);
newmouse(&MouseX,&MouseY,&press);
                                                                        bar(x_start,y_start,x_start+x,y_start+y);
           press_start(&bk);
                                                                        setcolor(BROWN);
      }
                                                                        setlinestyle(0, 0, 1);
      clrmous(MouseX,MouseY);
                                                                        line(50, 120, 50 + x, 120);
                                                                        line(50, 120 + y, 50 + x, 120 + y);
      warehouse list(h->here);
      delay(500);
                                                                        for (i = 50; i \le 50 + x; i += 25)
      while(1)
      {
                                                                             int temp = 0;
           if(re!=0)
                                                                             for (j = 120; j \le 120 + y; j += 1)
                                                                             {
           {
                break;
                                                                                  temp = rand() \% 2;
           }
                                                                                  line(i + temp, j, i + temp, j);
                                                                             }
newmouse(&MouseX,&MouseY,&press);
                                                                        }
                                                                        for (i = 0; i < 5000; i++)
           press_warelist(&re);
      }
      h->here[k].total[co_type]+=count;
                                                                             int x_{temp} = 0, y_{temp} = 0;
      if(h->here[k].total[co_type]<0)
                                                                             x_{temp} = rand() \% x;
      {
                                                                             y_temp = rand() % y;
           h->here[k].total[co_type]=ware_full;
                                                                             line(50 + x_{temp}, 120 + y_{temp}, 50 +
                                                                 x temp, 120 + y temp);
      mode=1;
                                                                       }
      return;
                                                                      }
    }
    // moudules which skip the video and show
                                                                      void select02(struct Parameter *abc)
```

```
int x[point_max], y[point_max], flag = 0,
                                                                    if (mouse_press(0, 0, 40, 30) == 2 ||
back = 0,i;
                                                              mouse press(200, 70, 280, 115) == 2 ||
      cleardevice();
                                                              mouse_press(320, 70, 400, 115) == 2)
      setbkcolor(WHITE);
                                                                    {
      setcolor(GREEN);
                                                                         MouseS = 1;
      setlinestyle(0, 0, 3);
                                                                    if (mouse\_press(0, 0, 40, 30) == 1)
      rectangle(x_start, y_start, x_start + x_max,
y_start + y_max);
     puthz(170, 30, "请依次在框内选点", 32, 32,
                                                                         mode=0;
BLUE);
      setfillstyle(1, GREEN);
                                                                    if (mouse press(200, 70, 280, 115) == 1)
      bar(200, 70, 280, 115);
                                                                    {
      puthz(207, 76, "开始", 32, 32, WHITE);
                                                                         pick_points(x, y, flag);
      setfillstyle(1, RED);
                                                                         return;
      bar(320, 70, 400, 115);
                                                                    }
      puthz(327, 76, "完成", 32, 32, WHITE);
                                                                    //
      settextstyle(3, 0, 4);
                                                              if(mouse_press(200,70,280,115)==1&&(*flag)!=0)
      quit();
                                                                    //{
                                                                    //
                                                                         pick_points(x,y,flag);
      while (1)
                                                                    //
                                                                         return;
                                                                    // }
      {
          if (back != 0)
                                                                    if (mouse_press(320, 70, 400, 115) == 1)
               break;
                                                                         //init_field02(x, y, flag);
                                                                         *back = 1;
          newmouse(&MouseX,
                                       &MouseY,
                                                                    }
&press);
                                                                   }
          press_select02(x, y, &flag, &back);
          delay(20);
                                                                   void pick_wait()
      }
                                                                   {
      abc->lenxy=flag;
                                                                    int i;
      for (i=0;i<flag;i++)
                                                                    for (i = 0; i < 30; i++)
      {
                                                                    {
          abc->x[i]=x[i];
                                                                         newmouse(&MouseX,
                                                                                                     &MouseY,
          abc->y[i]=y[i];
                                                              &press);
      }
                                                                         delay(10);
    }
                                                                    }
                                                                   }
    void press_select02(int *x, int *y, int *flag,
int *back)
                                                                   // void pick_start(int *x,int *y)
    {
                                                                   //{
      if (mouse_press(0, 0, 40, 30) == 0 ||
                                                                         while(1)
                                                                   //
mouse_press(200, 70, 280, 115) == 0 ||
                                                                  //
                                                                         {
mouse_press(320, 70, 400, 115) == 0)
                                                                   //
      {
                                                               newmouse(&MouseX,&MouseY,&press);
          MouseS = 0;
                                                                  //
```

```
if(mouse_press(x_start,y_start,x_start+x_max,y_
                                                              y_start + y_max) == 1
start+y_max)==1)
               {
                                                                              x[*flag] = MouseX;
    //
    //
                    *x=MouseX;
                                                                              y[*flag] = MouseY;
                    *v=MouseY;
    //
                                                                              clrmous(MouseX, MouseY);
    //
                    break;
                                                                              setfillstyle(1, RED);
    //
               }
                                                                              bar(x[*flag] - 5, y[*flag] - 5,
    //
                delay(20);
                                                              x[*flag] + 5, y[*flag] + 5);
    //
          }
                                                                              setcolor(BLUE);
    //
                                                                              line(x[(*flag) - 1], y[(*flag) - 1],
           clrmous(MouseX,MouseY);
    //
           setfillstyle(1,RED);
                                                              x[*flag], y[*flag]);
    //
           bar((*x)-5,(*y)-5,(*x)+5,(*y)+5);
                                                                              (*flag)++;
    //
          setfillstyle(1,GREEN);
                                                                         }
    //
           bar(200,70,280,115);
                                                                         if (mouse press(0, 0, 40, 30) == 0 | |
    //
           puthz(207,76,"继续",32,32,WHITE);
                                                              mouse press(320, 70, 400, 115) == 0)
    //
           return;
                                                                         {
    //}
                                                                              MouseS = 0;
                                                                         if (mouse press(0, 0, 40, 30) == 2 ||
    void pick points(int *x, int *y, int *flag)
                                                              mouse_press(320, 70, 400, 115) == 2)
      clrmous(MouseX, MouseY);
                                                                         {
      setfillstyle(1, WHITE);
                                                                              MouseS = 1;
      bar(200, 70, 280, 115);
                                                                         }
      // mouseinit();
                                                                         if (mouse\_press(0, 0, 40, 30) == 1)
      while (*flag == 0)
      {
                                                                              mode=0;
           newmouse(&MouseX,
                                       &MouseY,
                                                                              return;
&press);
                                                                         }
                                                                         //
                (mouse_press(x_start,
                                         y_start,
x_start + x_max, y_start + y_max) == 1
                                                              if(mouse_press(200,70,280,115)==1&&(*flag)==0)
                                                                         //{
           {
                *x = MouseX;
                                                                         //
                                                                              pick_points(x,y,flag);
                *y = MouseY;
                                                                         //
                                                                              return;
               clrmous(MouseX, MouseY);
                                                                         //}
               setfillstyle(1, GREEN);
                                                                         //
               bar(x[*flag] - 5, y[*flag] - 5,
                                                              if(mouse_press(200,70,280,115)==1&&(*flag)!=0)
x[*flag] + 5, y[*flag] + 5);
                                                                         //{
               (*flag)++;
                                                                         //
                                                                              pick_points(x,y,flag);
          }
                                                                         //
                                                                              return;
           delay(20);
                                                                         //}
                                                                         if (mouse press(320, 70, 400, 115) ==
      }
      while ((*flag) < point max)
                                                              1)
                                                                         {
      {
           newmouse(&MouseX,
                                       &MouseY,
                                                                              break;
&press);
                                                                              // init_field02(x,y,flag);
           if (MouseX != x[(*flag) - 1] &&
mouse_press(x_start, y_start, x_start + x_max,
                                                                         delay(20);
```

```
}
                                                                            int x_r = rand() \% (xy_m[2] - xy_m[0]),
      return;
                                                                 y_r = rand() \% (xy_m[3] - xy_m[1]);
    }
                                                                            line(xy_m[0] + x_r, xy_m[1] + y_r,
                                                                 xy_m[0] + x_r, xy_m[1] + y_r);
    // void pick points(int *x,int *y,int *flag)
    //{
                                                                       return;
    //
           while((*flag)<point max)</pre>
                                                                      }
    //
           {
    //
                                                                      /*void pick_points(int *x,int *y)
 newmouse(&MouseX,&MouseY,&press);
    //
                                                                       int xn,yn,i=1;
 if(mouse_press(x_start,y_start,x_start+x_max,y_
                                                                       setfillstyle(1,RED);
start+y_max)==1)
                                                                       setlinestyle(0,0,1);
    //
                {
                                                                       setcolor(BLUE);
    //
                     x[*flag]=MouseX;
                                                                       while (xn < x[0] - 5\&\&xn > x[0] + 5\&\&yn < y[0] - 5\&
    //
                     y[*flag]=MouseY;
                                                                 yn>x[0]+5
    //
                     break;
                                                                       {
    //
                                                                            pick_wait();
    //
                delay(20);
                                                                            bar(xn-5,yn-5,xn+5,yn+5);
    //
                                                                            x[i]=xn;
    //
           clrmous(MouseX,MouseY);
                                                                            y[i]=yn;
    //
           setfillstyle(1,RED);
                                                                            i++;
    //
                                                                            while(1)
 bar(x[*flag]-5,y[*flag]-5,x[*flag]+5,y[*flag]+5);
    //
           (*flag)++;
    //
           return;
                                                                  newmouse(&MouseX,&MouseY,&press);
    //}
                                                                  if(mouse_press(x_start,y_start,x_start+x_max,y_
    void draw_points(int *arr,int *flag,long int
                                                                 start+y_max)==1)
*xy_m)
                                                                                 {
                                                                                      xn=MouseX;
      int i=0;
                                                                                      yn=MouseY;
      setfillstyle(1, WHITE);
                                                                                      break;
      fillpoly(*flag, arr);
                                                                                 }
      setcolor(BROWN);
                                                                                 delay(15);
      for (i = xy_m[0]; i \le xy_m[2]; i += 25)
                                                                            }
      {
                                                                            line(x[i],y[i],xn,yn);
           int temp = 0, j;
           for (j = xy_m[1]; j \le xy_m[3]; j ++)
                                                                       return;
           {
                                                                      }*/
                temp = rand() \% 2;
                line(i + temp, j, i + temp, j);
                                                                      void select03(struct Parameter *abc)
           }
                                                                      {
                                                                       int
                                                                                              x[dense_points_max],
      for (i = 0; i < ((xy_m[2] - xy_m[0]) * (xy_m[3])
                                                                 y[dense_points_max], flag = 0, back=0,i;
- xy_m[1])) * 0.05; i++)
                                                                       cleardevice();
      {
                                                                       setbkcolor(WHITE);
```

```
setcolor(GREEN);
                                                                     {
      setlinestyle(0, 0, 3);
                                                                          MouseS=1;
      rectangle(x_start, y_start, x_start + x_max,
                                                                     if(mouse\_press(0,0,40,30)==1)
y_start + y_max);
     puthz(170, 30, "请缓慢移动鼠标勾勒图形",
32, 32, BLUE);
                                                                          mode=0;
      setfillstyle(1, GREEN);
                                                                          return;
      bar(200, 70, 280, 115);
      puthz(207, 76, "开始", 32, 32, WHITE);
                                                                     if(mouse_press(200,70,280,115)==1&&(*fla
      setfillstyle(1, RED);
                                                               g) = = 0
      bar(320, 70, 400, 115);
                                                                     {
      puthz(327, 76, "完成", 32, 32, WHITE);
                                                                          dense_pick(x,y,flag);
      settextstyle(3, 0, 4);
                                                                          return;
      quit();
                                                                     if(mouse_press(320,70,400,115)==1)
      setfillstyle(1, RED);
      while (1)
                                                                          //init_field03(x,y,flag,0);
                                                                          *back=1;
      {
           if (back != 0)
                                                                          return;
           {
                                                                    }
                break;
                                                                   }
           newmouse(&MouseX,
                                       &MouseY,
                                                                    void dense_pick(int *x, int *y, int *flag)
&press);
           press_select03(x, y, &flag,&back);
                                                                     clrmous(MouseX, MouseY);
           delay(20);
                                                                     bar(200, 70, 280, 115);
      }
                                                                     setlinestyle(0, 0, 1);
      abc->lenxy=flag;
                                                                     setcolor(BLUE);
      for (i=0;i<flag;i++)
                                                                     while ((*flag) == 0)
      {
           abc->x[i]=x[i];
                                                                          newmouse(&MouseX,
                                                                                                       &MouseY,
           abc->y[i]=y[i];
                                                               &press);
      }
                                                                               (mouse_press(x_start,
                                                                                                         y_start,
    }
                                                               x_start + x_max, y_start + y_max) == 1
    void press_select03(int *x,int *y,int *flag,int
                                                                               *x = MouseX;
*back)
                                                                               *y = MouseY;
                                                                               clrmous(MouseX, MouseY);
      if(mouse_press(0,0,40,30)==0||(mouse_pr
                                                                               setfillstyle(1, GREEN);
ess(200,70,280,115)==0&&(*flag==0))||mouse_p
                                                                               bar(x[*flag] - 5, y[*flag] - 5,
ress(320,70,400,115)==0)
                                                               x[*flag] + 5, y[*flag] + 5);
      {
                                                                               (*flag)++;
           MouseS=0;
                                                                               break;
      if(mouse_press(0,0,40,30)==2||(mouse_pr
                                                                          delay(20);
ess(200,70,280,115)==2&&(*flag==0))||mouse_p
ress(320,70,400,115)==2)
                                                                     pick_wait();
```

```
setfillstyle(1, RED);
      while ((*flag) < dense_points_max)
                                                                        for (i = 0; i < ((xy_m[2] - xy_m[0]) * (xy_m[3])
                                                                  - xy_m[1])) * 0.05; i++)
      {
           if (((*flag) >= 5 \&\& abs(MouseX - x[0]))
                                                                        {
<= 3 && abs(MouseY - y[0]) <= 3) || MouseX <=
                                                                             int x_r = rand() \% (xy_m[2] - xy_m[0]),
x start || MouseX >= x start + x max || MouseY
                                                                  y_r = rand() \% (xy_m[3] - xy_m[1]);
<= y_start || MouseY >= y_start + y_max)
                                                                             line(xy_m[0] + x_r, xy_m[1] + y_r,
                                                                  xy_m[0] + x_r, xy_m[1] + y_r);
                line(x[0], y[0], x[(*flag) - 1],
                                                                        for (i = xy m[0]; i \le xy m[2]; i += 25)
y[(*flag) - 1]);
                return;
           }
                                                                             int j = 0;
           if ((MouseX - x[(*flag) - 1]) * (MouseX -
                                                                             for (j = xy_m[1]; j \le xy_m[3]; j++)
x[(*flag) - 1]) >= 4 && (MouseY - y[(*flag) - 1]) *
(MouseY - y[(*flag) - 1]) >= 4)
                                                                                  int temp = rand() \% 2;
                                                                                  line(i + temp, j, i + temp, j);
           {
                                                                             }
                x[*flag] = MouseX;
                y[*flag] = MouseY;
                                                                        }
                clrmous(MouseX, MouseY);
                                                                        return;
                setfillstyle(1, RED);
                                                                       }
                bar(x[*flag] - 5, y[*flag] - 5,
                                                                       void init field02(int *x,int *y,int *flag,int
x[*flag] + 5, y[*flag] + 5);
                line(x[(*flag) - 1], y[(*flag) - 1],
                                                                  type,int time)
x[(*flag)], y[(*flag)]);
                                                                       {
                (*flag)++;
                                                                        // The meaning of elements in xy m:
                                                                        //[0]:minest of x,[1]:minest of y,[2]:largest
           }
                                                                  of x,[3]:largest of y
           pick_wait();
      }
                                                                        long
                                                                                                                   int
                                                                  xy_m[4]={x_start+x_max,y_start+y_max,x_start,y
      return;
    }
                                                                  _start};
                                                                                        i,arr[point max
     void dense_draw_points(int *arr, int *flag,
                                                                  2],num=0,xy[2],des_x[tracktor_num_max],des_y[t
long int *xy_m)
                                                                  racktor_num_max];
                                                                        for (i = 0; i < (*flag); i++)
     {
      int i;
      setfillstyle(1, WHITE);
                                                                             if (x[i] < xy_m[0])
      setcolor(WHITE);
                                                                             {
      fillpoly(*(flag), arr);
                                                                                  xy_m[0] = x[i];
      setcolor(BROWN);
      for (i = xy_m[0]; i \le xy_m[2]; i += 25)
                                                                             if (y[i] < xy_m[1] & y[i]!=0
      {
           int temp = 0, j;
                                                                                  xy m[1] = y[i];
           for (j = xy_m[1]; j \le xy_m[3]; j += 2)
           {
                                                                             if (x[i] > xy_m[2])
                temp = rand() \% 2;
                line(i + temp, j, i + temp, j);
                                                                                  xy_m[2] = x[i];
           }
                                                                             }
```

```
if (y[i] > xy_m[3])
                                                                         clrmous(MouseX,MouseY);
                                                                         draw_setoff(xy);
          {
               xy_m[3] = y[i];
          }
                                                               dense_init_tracktor01(x,y,flag,xy_m,num,des_x,
          arr[2 * i] = x[i];
                                                              des y);
          arr[2 * i + 1] = y[i];
      }
                                                               tracktor_return(xy[0],xy[1],xy_m[0]+(xy_m[2]-xy
      for(i=0;i<num;i++)
                                                              _m[0])/num,xy_m[1],(xy_m[2]-xy_m[0])/num,nu
                                                              m);
          des x[i]=600;
          des_y[i]=0;
                                                               picker_anime(xy[0],xy[1],des_x,des_y,(xy_m[2]-x
      }
                                                              y m[0])/num,num);
      select_setoff02(xy,xy_m[0],xy_m[1],xy_m[2
                                                                    }
],xy_m[3]);
                                                                    else
      draw simu01(time);
                                                                    {
      // setfillstyle(1,BROWN);
      // setlinestyle(0,0,3);
                                                               tracktor\_set\_off0(xy[0],xy[1],xy\_m[0],xy\_m[3],(x
                                                              y_m[2]-xy_m[0])/num,num);
bar(x start-5,y start-5,x start+x max+5,y start+y
                                                                         clrmous(MouseX,MouseY);
_max+5);
                                                                         draw_setoff(xy);
      draw points(arr,flag,xy m);
      clrmous(MouseX,MouseY);
                                                               dense_init_tracktor02(x,y,flag,xy_m,num,des_x,
      draw_setoff(xy);
                                                              des_y);
      //setcolor(GREEN);
      //rectangle(xy_m[0],xy_m[1],xy_m[2],xy_m
                                                               tracktor_return0(xy[0],xy[1],xy_m[0]+(xy_m[2]-x
[3]);
                                                              y_m[0])/num,xy_m[1],(xy_m[2]-xy_m[0])/num,nu
      // setcolor(BROWN);
                                                              m);
for(i=0;i<((xy_m[2]-xy_m[0])*(xy_m[3]-xy_m[1]))/
                                                               picker_anime(xy[0],xy[1],des_x,des_y,(xy_m[2]-x
20;i++)
                                                              y_m[0])/num,num);
      //{
                                                                    }
      // int
                                                                   }
x_r=rand()\%(xy_m[2]-xy_m[0]),y_r=rand()\%(xy_m[2]-xy_m[0])
[3]-xy_m[1]);
                                                                   void init field03(int *x,int *y,int *flag,int
     //
                                                              type,int time)
 line(xy_m[0]+x_r,xy_m[1]+y_r,xy_m[0]+x_r,xy_m
[1]+y_r);
                                                                  {
                                                                    // The meaning of elements in xy m:
      //}
      num=time/tra_time;
                                                                    //[0]:minest of x,[1]:minest of y,[2]:largest
      if(num==0)
                                                              of x,[3]:largest of y
          num=1;
      //delay(1000);
                                                              xy m[4]={x start+x max,y start+y max,x start,y
      if(type==0)
                                                              start};
      {
                                                                           arr[dense_points_max
                                                                                                            2],
                                                              i,num=0,xy[2],des_x[tracktor_num_max],des_y[tr
 tracktor_set_off(xy[0],xy[1],xy_m[0],xy_m[3],(xy
                                                              acktor num max];
_m[2]-xy_m[0])/num,num);
                                                                    num=time/tra time;
```

```
// int
      if(num==0)
                                                              x_r=rand()%(xy_m[2]-xy_m[0]),y_r=rand()%(xy_m
           num=1;
      for (i = 0; i < (*flag); i++)
                                                              [3]-xy_m[1]);
                                                                     //
      {
           if (x[i] < xy_m[0])
                                                                line(xy_m[0]+x_r,xy_m[1]+y_r,xy_m[0]+x_r,xy_m
                                                              [1]+y_r);
               xy_m[0] = x[i];
                                                                    //}
                                                                    //delay(1000);
           }
           if (y[i] < xy_m[1])
                                                                     if(type==0)
           {
                                                                     {
               xy_m[1] = y[i];
           }
                                                                tracktor_set_off(xy[0],xy[1],xy_m[0],xy_m[3],(xy
           if (x[i] > xy_m[2])
                                                              _m[2]-xy_m[0])/num,num);
                                                                          clrmous(MouseX,MouseY);
           {
               xy_m[2] = x[i];
                                                                          draw_setoff(xy);
           }
           if (y[i] > xy_m[3])
                                                                dense_init_tracktor01(x,y,flag,xy_m,num,des_x,
                                                              des_y);
           {
               xy_m[3] = y[i];
           }
                                                                tracktor_return(xy[0],xy[1],xy_m[0]+(xy_m[2]-xy
           arr[2 * i] = x[i];
                                                              m[0])/num,xy m[1],(xy m[2]-xy m[0])/num,nu
           arr[2 * i + 1] = y[i];
                                                              m);
      }
      for(i=0;i<num;i++)
                                                                picker_anime(xy[0],xy[1],des_x,des_y,(xy_m[2]-x
                                                              y_m[0])/num,num);
      {
                                                                     }
           des x[i]=600;
           des_y[i]=0;
                                                                     else
      }
                                                                     {
      // setfillstyle(1,BROWN);
      // setlinestyle(0,0,3);
                                                                tracktor\_set\_off0(xy[0],xy[1],xy\_m[0],xy\_m[3],(x
                                                              y_m[2]-xy_m[0])/num,num);
bar(x_start-5,y_start-5,x_start+x_max+5,y_start+y
                                                                         clrmous(MouseX,MouseY);
                                                                         draw_setoff(xy);
max+5);
      select_setoff02(xy,xy_m[0],xy_m[1],xy_m[2
],xy_m[3]);
                                                                dense init tracktor02(x,y,flag,xy m,num,des x,
      draw simu01(time);
                                                              des_y);
      dense_draw_points(arr,flag,xy_m);
      clrmous(MouseX,MouseY);
                                                                tracktor_return0(xy[0],xy[1],xy_m[0]+(xy_m[2]-x
                                                              y_m[0])/num,xy_m[1],(xy_m[2]-xy_m[0])/num,nu
      draw_setoff(xy);
      //setcolor(GREEN);
                                                              m);
      //rectangle(xy_m[0],xy_m[1],xy_m[2],xy_m
[3]);
                                                                picker anime(xy[0],xy[1],des x,des y,(xy m[2]-x
      // setcolor(BROWN);
                                                              y_m[0])/num,num);
                                                                    }
for(i=0;i<((xy_m[2]-xy_m[0])*(xy_m[3]-xy_m[1]))/
                                                                   }
20;i++)
      //{
                                                                   void dense init tracktor01(int *x, int *y, int
```

```
*flag,long int *xy_m, int num,int *des_x,int
                                                                                         {
*des_y)
                                                                                              tra_d[i][3]=y[k];
     {
                                                                                              tra_mark[i][3]=k;
      long int total=0;
                                                                                         }
                                                                                         break;
x_d,i,tra_d[tracktor_num_max][4],tra_mark[trackt
                                                                                    }
or_num_max][4];
                                                                                    else
      x_d=xy_m[2]-xy_m[0];
                                                                                    {
      x_d/=num;
                                                                                         k++;
      for(i=0;i<num;i++)
                                                                                    }
                                                                               }
      {
            int k=0;
                                                                          for(i=0;i<num;i++)
            tra_d[i][0]=xy_m[0]+i*x_d;
            tra_d[i][2]=xy_m[0]+(i+1)*x_d;
            tra_d[i][1]=xy_m[3];
            tra_d[i][3]=xy_m[1];
                                                                    if(tra\_d[i][1]>y[tra\_mark[i][1]-1]\&\&y[tra\_mark[i]
                                                                   [1]-1]!=0)
            while(k<(*flag))
                                                                               {
           {
                 int target=0;
                                                                    tra\_d[i][1] = (y[tra\_mark[i][1] - 1] + y[tra\_mark[i][1]]
 if(x[k] > = tra_d[i][0] \& x[k] < = tra_d[i][2])
                                                                   )/2-30;
                                                                               }
                 {
                      if(target==0)
                      {
                                                                    if(tra_d[i][1]>y[tra_mark[i][1]+1]&&y[tra_mark[i
                                                                   ][1]+1]!=0)
                           target=1;
                                                                               {
                      }
                      if(y[k] < tra\_d[i][1]) \\
                                                                    tra_d[i][1]=(y[tra_mark[i][1]+1]+y[tra_mark[i][1]
                           tra_d[i][1]=y[k];
                                                                   ])/2-30;
                           tra_mark[i][1]=k;
                                                                               }
                                                                               if(tra_d[i][3]<y[tra_mark[i][3]-1])
                      }
                      if(y[k]>tra_d[i][3])
                      {
                           tra_d[i][3]=y[k];
                                                                    tra_d[i][3]=(y[tra_mark[i][3]-1]+y[tra_mark[i][3]]
                           tra_mark[i][3]=k;
                                                                   )/2+30;
                      }
                                                                               }
                 }
                                                                               if(tra_d[i][3]<y[tra_mark[i][3]+1])
                                                                               {
 if(target!=0\&\&(x[k]< tra_d[i][0]||x[k]> tra_d[i][2])
)
                                                                    tra_d[i][3]=(y[tra_mark[i][3]+1]+y[tra_mark[i][3]
                 {
                                                                   ])/2+30;
                                                                               }
                      if(y[k]<tra_d[i][1])
                      {
                                                                          }
                           tra_d[i][1]=y[k];
                                                                         for(i=0;i<num;i++)
                           tra_mark[i][1]=k;
                      }
                                                                               total+=tra_d[i][1];
                      if(y[k]>tra_d[i][3])
                                                                               total+=tra_d[i][3];
```

```
}
                                                                                      if(y[k]<tra_d[i][1])
      // for(i=0;i<num;i++)
      //{
                                                                                           tra_d[i][1]=y[k];
      // int
                                                                                           tra_mark[i][1]=k;
temp=(total-tra_d[i][1]-tra_d[i][3])/((num-1)*2);
      // if(tra_d[i][3]<temp)
                                                                                      if(y[k]>tra_d[i][3])
      // {
                                                                                      {
      //
                                                                                           tra_d[i][3]=y[k];
 tra_d[i][3]=(xy_m[3]+tra_d[i][3])/2+30;
                                                                                           tra_mark[i][3]=k;
      // }
                                                                                      }
      // if(tra_d[i][1]>temp)
                                                                                 }
      // {
                                                                  if(target!=0\&\&(x[k]< tra_d[i][0]||x[k]> tra_d[i][2])
      //
 tra_d[i][1]=(xy_m[1]+tra_d[i][1])/2-30;
                                                                )
      // }
                                                                                 {
      //}
                                                                                      if(y[k] < tra_d[i][1])
      start_ainime03_01(tra_d,num,xy_m,des_x,
                                                                                      {
                                                                                           tra_d[i][1]=y[k];
des_y);
    }
                                                                                           tra_mark[i][1]=k;
                                                                                      }
    void dense init tracktor02(int *x,int *y,int
                                                                                      if(y[k]>tra_d[i][3])
*flag,long int* xy_m,int num,int *des_x,int
                                                                                      {
*des_y)
                                                                                           tra_d[i][3]=y[k];
    {
                                                                                           tra_mark[i][3]=k;
      long int total=0;
                                                                                      }
                                                                                      break;
      int
x_d,i,tra_d[tracktor_num_max][4],tra_mark[trackt
                                                                                 }
or_num_max][4];
                                                                                 else
                                                                                 {
      x_d=xy_m[2]-xy_m[0];
      x_d/=num;
                                                                                      k++;
      for(i=0;i<num;i++)
                                                                                 }
                                                                            }
           int k=0;
           tra_d[i][0]=xy_m[0]+i*x_d;
                                                                       for(i=0;i<num;i++)
           tra d[i][2]=xy m[0]+(i+1)*x d;
           tra_d[i][1]=xy_m[3];
           tra_d[i][3]=xy_m[1];
                                                                  if(tra_d[i][1]>y[tra_mark[i][1]-1]&&y[tra_mark[i]
           while(k<(*flag))
                                                                [1]-1]!=0)
           {
                                                                            {
                int target=0;
                                                                  tra_d[i][1]=(y[tra_mark[i][1]-1]+y[tra_mark[i][1]]
                                                                )/2-30;
 if(x[k] > = tra d[i][0] \& x[k] < = tra d[i][2])
                                                                            }
                     if(target==0)
                                                                  if(tra_d[i][1]>y[tra_mark[i][1]+1]&&y[tra_mark[i
                     {
                                                                ][1]+1]!=0)
                          target=1;
                     }
                                                                            {
```

```
y_p[tracktor_num_max],cal_time[tracktor_
 tra_d[i][1]=(y[tra_mark[i][1]+1]+y[tra_mark[i][1]
                                                                num_max],i;
1)/2-30;
                                                                      for(i=0;i<num;i++)
                                                                      {
           if(tra_d[i][3]<y[tra_mark[i][3]-1])
                                                                           type[i] = 0;
                                                                           cal_time[i]=0;
                                                                           x_p[i] = tra_d[i][0];
 tra_d[i][3]=(y[tra_mark[i][3]-1]+y[tra_mark[i][3]]
                                                                           y_p[i] = tra_d[i][3]-40;
)/2+30;
                                                                      }
                                                                      while (1)
           }
           if(tra_d[i][3]<y[tra_mark[i][3]+1])
                                                                      {
                                                                           int count=0,re=0;
 tra_d[i][3]=(y[tra_mark[i][3]+1]+y[tra_mark[i][3]
                                                                 newmouse(&MouseX,&MouseY,&press);
])/2+30;
           }
                                                                 re=pressed_anime(xy_m[0],xy_m[1],xy_m[2],xy_
      }
                                                                m[3]);
      for(i=0;i<num;i++)
                                                                           if(re!=0)
                                                                           {
                                                                                for(i=0;i<num;i++)
           total+=tra_d[i][1];
           total+=tra d[i][3];
                                                                                {
      }
                                                                                     if(type[i]==0)
      // for(i=0;i<num;i++)
      //{
                                                                                          earth_cover01(x_p[i],
      // int
                                                                y_p[i]);
temp=(total-tra d[i][1]-tra d[i][3])/((num-1)*2);
                                                                                          earth cover01(x p[i],
      // if(tra_d[i][3]<temp)
                                                                y_p[i] + 5);
      // {
                                                                                     if(type[i]==1)
      //
 tra_d[i][3]=(xy_m[3]+tra_d[i][3])/2+30;
                                                                                     {
      // if(tra_d[i][1]>temp)
                                                                 earth_cover02(x_p[i],y_p[i]);
      // {
      //
                                                                 earth_cover02(x_p[i]+7,y_p[i]);
 tra_d[i][1]=(xy_m[1]+tra_d[i][1])/2-30;
                                                                                     }
      // }
                                                                                }
      //}
                                                                                if(mode==0)
      start ainime03 02(tra d,num,xy m,des x,
des_y);
                                                                                     return;
    }
                                                                                }
                                                                                break;
     void start ainime03 01(int (*tra d)[4], int
num,long int *xy_m,int *des_x,int *des_y)
                                                                           for(i=0;i<num;i++)
     {
      int
                                                                                if (type[i] == 4)
type[tracktor_num_max],x_p[tracktor_num_max]
                                                                                {
\backslash
                                                                                     count++;
```

```
}
                                                                  y_p[i]);
           }
                                                                                       y_p[i]--;
           for (i = 0; i < num; i++)
                                                                                       if (y_p[i] <= tra_d[i][1])
           {
                if(cal_time[i]>=0)
                                                                                            earth_fill03(x_p[i], y_p[i]
                                                                  - 7);
                     cal time[i]++;
                                                                                             type[i] = 1;
                }
                                                                                             x_p[i] += 25;
                                                                                       }
 if(cal_time[i]>0&&cal_time[i]>=pick_bar)
                                                                                        delay(delaytime / (num -
                {
                                                                  count));
                      cal_time[i]=-1;
                                                                                        continue;
                     if(type[i]==0)
                                                                                  }
                                                                                  else if (type[i] == 1)
                     {
                                                                                  {
 draw_copak(x_p[i],y_p[i]+tracktor_l+co_pak_w);
                                                                                        earth_fill02(x_p[i], y_p[i]);
                           des_x[i]=x_p[i];
                                                                                       init_tracktor01_b(x_p[i],
                                                                  y_p[i]);
 des_y[i]=y_p[i]+tracktor_l+co_pak_w;
                                                                                       y_p[i]++;
                                                                                       if (y_p[i] + 40 >= tra_d[i][3])
                     }
                      else
                     {
                                                                                             if
                                                                                                               25 >
                                                                                                 (x_p[i] +
                                                                  tra_d[i][2])
 draw_copak(x_p[i],y_p[i]-2*co_pak_w);
                                                                                             {
                           des_x[i]=x_p[i];
                                                                                                  type[i] = 3;
                                                                                            }
 des_y[i]=y_p[i]-2*co_pak_w;
                                                                                             else
                                                                                             {
                }
                                                                                                  earth_fill03(x_p[i],
                if ((type[i] != 0 \&\& y_p[i] - 40 >=
                                                                  y_p[i] - 1);
y_{t} = tra_{d}[i][2]
                                                                                                  earth_fill03(x_p[i],
                                                                  y_p[i] + 2);
                     if (type[i] != 4)
                                                                                                  type[i] = 2;
                     {
                                                                                                  x_p[i] += 25;
                                                                                            }
                           init_tracktor01_f(x_p[i]
- 25, y_p[i]);
                                                                                       }
                           type[i] = 4;
                                                                                        delay(delaytime / (num -
                     }
                                                                  count));
                     else
                                                                                        continue;
                     {
                                                                                  }
                           continue;
                                                                                  else
                     }
                                                                                  {
                                                                                        type[i]=2;
                else if (type[i] == 0)
                                                                                        earth_fill01(x_p[i], y_p[i]);
                                                                                       init_tracktor01_f(x_p[i],
                      earth_fill01(x_p[i], y_p[i]);
                                                                  y_p[i]);
                     init_tracktor01_f(x_p[i],
                                                                                       y_p[i]--;
```

```
if (y_p[i] <= tra_d[i][1])
                                                                 type[tracktor_num_max],x_p[tracktor_num_max]
                                                                 \backslash
                          earth_fill03(x_p[i], y_p[i]
                                                                       y_p[tracktor_num_max],i,cal_time[tracktor
- 7);
                                                                 _num_max];
                                                                       for(i=0;i<num;i++)
                          type[i] = 1;
                          x_p[i] += 25;
                                                                       {
                     }
                                                                            type[i] = 0;
                     delay(delaytime / (num -
                                                                            cal_time[i]=0;
count));
                                                                            x_p[i] = tra_d[i][0];
                     continue;
                                                                            y_p[i] = tra_d[i][3]-40;
                }
                                                                       }
                // else if (type[i] = 3)
                                                                       while (1)
                //{
                                                                       {
                     earth_fill02(x_p[i], y_p[i]);
                                                                            int count=0,re=0;
                //
                //
                     init_tracktor01_b(x_p[i],
y_p[i]);
                                                                  newmouse(&MouseX,&MouseY,&press);
                //
                     y_p[i]++;
                //}
                                                                  re=pressed_anime(xy_m[0],xy_m[1],xy_m[2],xy_
           }
                                                                 m[3]);
           for(i=0;i<num;i++)
                                                                            if(re!=0)
           {
                                                                            {
                draw_copak(des_x[i],des_y[i]);
                                                                                 for(i=0;i<num;i++)
           }
                                                                                 {
           if (count >= num)
                                                                                      if(type[i]==0)
           {
                break;
                                                                                            earth_cover01(x_p[i],
           }
                                                                 y_p[i]);
      }
                                                                                           earth_cover01(x_p[i],
      for(i=0;i<num;i++)
                                                                 y_p[i] + 5);
      {
                                                                                      }
           earth_fill03(x_p[i]-25,y_p[i]);
                                                                                      if(type[i]==1)
           earth_fill03(x_p[i]-25,y_p[i]-6);
           if(cal_time[i]>=0)
           {
                                                                  earth_cover02(x_p[i],y_p[i]);
                des x[i]=0;
                                                                  earth_cover02(x_p[i]+7,y_p[i]);
           }
           // else
                                                                                      }
           //{
                                                                                 }
           //
                draw_copak(des_x[i],des_y[i]);
                                                                                 if(mode==0)
           //}
                                                                                 {
      }
                                                                                       return;
     }
                                                                                 }
                                                                                 break;
     void start_ainime03_02(int (*tra_d)[4], int
num,long int *xy_m,int *des_x,int *des_y)
                                                                            for(i=0;i<num;i++)
     {
                                                                            {
      int
                                                                                 if (type[i] == 4)
```

```
{
                                                                                         earth_fill01(x_p[i], y_p[i]);
                      count++;
                                                                                         init_tracktor02_f(x_p[i],
                 }
                                                                   y_p[i]);
           }
                                                                                         y_p[i]--;
            for (i = 0; i < num; i++)
                                                                                         if (y_p[i] <= tra_d[i][1])
                 if(cal_time[i]>=0)
                                                                                              earth_fill03(x_p[i], y_p[i]
                                                                   - 7);
                      cal_time[i]++;
                                                                                              type[i] = 1;
                                                                                              x_p[i] += 25;
                 }
                                                                                         }
                                                                                         delay(delaytime / (num -
 if(cal_time[i]>0&&cal_time[i]>=pick_bar)
                                                                   count));
                      cal_time[i]=-1;
                                                                                         continue;
                      if(type[i]==0)
                                                                                    }
                      {
                                                                                    else if (type[i] == 1)
                                                                                    {
 draw_copak(x_p[i],y_p[i]+tracktor_l+co_pak_w);
                                                                                         earth_fill02(x_p[i], y_p[i]);
                           des_x[i]=x_p[i];
                                                                                         init_tracktor02_b(x_p[i],
                                                                   y_p[i]);
 des_y[i]=y_p[i]+tracktor_l+co_pak_w;
                                                                                         y p[i]++;
                                                                                         if (y_p[i] + 40 >= tra_d[i][3])
                      }
                      else
                      {
                                                                                                  (x_p[i] + 25 >
                                                                   tra_d[i][2])
 draw_copak(x_p[i],y_p[i]-2*co_pak_w);
                                                                                              {
                           des_x[i]=x_p[i];
                                                                                                    type[i] = 3;
                                                                                              }
                                                                                              else
 des_y[i]=y_p[i]-2*co_pak_w;
                      }
                                                                                              {
                                                                                                    earth_fill03(x_p[i],
                 if ((type[i] != 0 \&\& y_p[i] - 40 >=
                                                                   y_p[i] - 1);
y_start + y_max) \mid \mid x_p[i] >= tra_d[i][2]
                                                                                                    earth_fill03(x_p[i],
                                                                   y_p[i] + 2);
                 {
                      if (type[i] != 4)
                                                                                                    type[i] = 2;
                                                                                                    x_p[i] += 25;
                      {
                                                                                              }
                           init_tracktor02_f(x_p[i]
- 25, y_p[i]);
                           type[i] = 4;
                                                                                         delay(delaytime / (num -
                      }
                                                                   count));
                      else
                                                                                         continue;
                      {
                                                                                    }
                           continue;
                                                                                    else
                      }
                                                                                    {
                 }
                                                                                         type[i] = 2;
                 else if (type[i] == 0)
                                                                                         earth_fill01(x_p[i], y_p[i]);
                 {
                                                                                         init_tracktor02_f(x_p[i],
```

```
y_p[i]);
                                                                        void circle_field(long int r)
                      y_p[i]--;
                      if (y_p[i] <= tra_d[i][1])
                                                                        {
                                                                          long int x0 = (2 * x_start + x_max) / 2, y0 =
                      {
                          earth_fill03(x_p[i], y_p[i]
                                                                   (2 * y_start + y_max) / 2;
- 7);
                                                                          int i;
                                                                          setfillstyle(1,WHITE);
                           type[i] = 1;
                           x_p[i] += 25;
                                                                          fillellipse(x0,y0,r,r);
                      }
                                                                          setfillstyle(1,BROWN);
                      delay(delaytime / (num -
                                                                          bar(x_start,y_start,x_start+x_max,y_start+y
count));
                                                                   _max);
                      continue;
                                                                          if(r>=y_max/2-3)
                 }
                                                                          {
                 // else if (type[i] = 3)
                                                                               r = y max / 2 - 3;
                 //{
                      earth_fill02(x_p[i], y_p[i]);
                                                                          setfillstyle(1, WHITE);
                 //
                 //
                      init_tracktor02_b(x_p[i],
                                                                          fillellipse(x0, y0, r, r);
                                                                          setcolor(BROWN);
y_p[i]);
                                                                          for (i = 0; i < 4 * r * r / 20; i++)
                 //
                      y_p[i]++;
                 //}
            }
                                                                               int x r = rand() \% (2 * r), y r = rand() \%
            for(i=0;i<num;i++)
                                                                   (2 * r);
            {
                                                                               line(x0 - r + x_r, y0 - r + y_r, x0 - r + x_r,
                 draw_copak(des_x[i],des_y[i]);
                                                                   y0 - r + y_r);
            if (count >= num)
                                                                          for (i = x0 - r; i \le x0 + r; i += 25)
            {
                                                                          {
                 break;
                                                                               int j = 0;
            }
                                                                               for (j = y0 - r; j \le y0 + r; j++)
      }
                                                                               {
      for(i=0;i<num;i++)
                                                                                    int temp = rand() % 2;
                                                                                    line(i + temp, j, i + temp, j);
                                                                               }
            earth_fill03(x_p[i]-25,y_p[i]);
            earth_fill03(x_p[i]-25,y_p[i]-6);
                                                                          }
            if(cal time[i]>=0)
                                                                        }
            {
                 des_x[i]=0;
                                                                        void
                                                                                 cal_tracktor_circle(int
                                                                                                            type,double
                                                                   space,int time)
            }
           // else
                                                                        {
            //{
                                                                          long
                                                                                                                      int
            //
                 draw_copak(des_x[i],des_y[i]);
                                                                   x0=(2*x_start+x_max)/2,y0=(2*y_start+y_max)/2,
            //}
      }
                                                                          int
      // if(cal_time[0]>=0)
                                                                   tra_d[tracktor_num_max][4],i,d,num,xy[2],des_x[
      //{
                                                                   tracktor_num_max],des_y[tracktor_num_max];
           des_x[0]=0;
                                                                          r=sqrt(space/3.1415926)*10;
      //}
                                                                          num=time/tra_time;
```

```
if(num==0)
                                                                   tra_d[i][0]=2*x0-tra_d[num-i-1][2];
           num=1;
                                                                        //
      if(r>=y_max/2-3)
                                                                   tra_d[i][2]=2*x0-tra_d[num-i-1][0];
      {
                                                                        //
                                                                                  tra_d[i][1]=tra_d[num-i-1][1];
                                                                        //
                                                                                  tra d[i][3]=tra d[num-i-1][3];
           r = y max / 2 - 3;
                                                                        // }
      }
      for(i=0;i<num;i++)
                                                                        //}
      {
                                                                        // else
           des_x[i]=600;
                                                                        //{
           des y[i]=0;
                                                                             for(i=0;i<num/2;i++)
      }
                                                                        //
                                                                             {
      d = 2 * r / num;
                                                                        //
                                                                                  tra_d[i][0]=x0-r+i*d;
      for (i = 0; i < num; i++)
                                                                        //
                                                                                  tra_d[i][2]=tra_d[i][0]+d;
                                                                        //
           tra d[i][0] = x0 - r + i * d;
                                                                   tra_d[i][1]=y0-sqrt(r*r-(r-d*(i+1))*(r-d*(i+1)));
           tra_d[i][2] = tra_d[i][0] + d;
                                                                        //
                                                                                  tra_d[i][3]=2*y0-tra_d[i][1];
           if (tra_d[i][0] \le x0)
                                                                        //
                                                                             }
                                                                             tra_d[i][1]=y0-r;
           {
                                                                        //
                tra d[i][1] = y0 - sqrt(r * r - (r - d *
                                                                        // tra_d[i][3]=y0+r;
(i + 1)) * (r - d * (i + 1)));
                                                                        //
                                                                             for(i=num/2+1;i<num;i++)
           }
                                                                        //
                                                                             {
           else
                                                                        //
           {
                                                                   tra_d[i][0]=2*x0-tra_d[num-i-1][2];
                tra d[i][1] = y0 - sqrt(r * r - (r - d *
                                                                        //
i) * (r - d * i));
                                                                   tra_d[i][2]=2*x0-tra_d[num-i-1][0];
                                                                                  tra d[i][1]=tra d[num-i-1][1];
           }
                                                                        //
           tra_d[i][3] = 2 * y0 - tra_d[i][1];
                                                                        //
                                                                                  tra_d[i][3]=tra_d[num-i-1][3];
      }
                                                                        // }
                                                                        //}
      tra_d[num / 2 - 1][1] = y0 - r;
      tra_d[num / 2 - 1][3] = y0 + r;
                                                                        select_setoff02(xy,x0-r,y0-r,x0+r,y0+r);
      tra_d[num / 2][1] = y0 - r;
                                                                        draw simu01(time);
      tra_d[num / 2][3] = y0 + r;
                                                                        circle field(r);
      // if(num%2==0)
                                                                        clrmous(MouseX,MouseY);
      //{
                                                                        draw_setoff(xy);
                                                                        if(type==0)
      //
           for(i=0;i<num/2;i++)
      //
          {
                                                                        {
      //
                tra_d[i][0]=x0-r+i*d;
      //
                tra_d[i][2]=tra_d[i][0]+d;
                                                                   tracktor set off(xy[0],xy[1],x0-r,y0+r,2*r/num,n
      //
                                                                 um);
 tra_d[i][1]=y0-sqrt(r*r-(r-d*(i+1))*(r-d*(i+1)));
                                                                             clrmous(MouseX,MouseY);
                                                                             draw setoff(xy);
      //
                tra_d[i][3]=2*y0-tra_d[i][1];
      // }
      // tra_d[i-1][1]=y0-r;
                                                                   start_ainime04_01(tra_d,num,des_x,des_y);
      // tra_d[i-1][3]=y0+r;
                                                                             for(i=0;i<num;i++)
      // for(i=num/2;i<num;i++)</pre>
                                                                             {
      //
          {
      //
                                                                   earth fill03(x0-r+2*r/num+i*2*r/num,y0-r);
```

```
cal_time[i]=0;
 earth_fill03(x0-r+2*r/num+i*2*r/num,y0-r-6);
                                                                         x_p[i]=tra_d[i][0];
           }
                                                                         y_p[i]=y_start+y_max-40;
                                                                    }
                                                                    while(1)
 tracktor return(xy[0],xy[1],x0-r+2*r/num,y0-r,2*
r/num,num);
                                                                    {
                                                                         int count=0,re=0;
 picker_anime(xy[0],xy[1],des_x,des_y,2*r/num,n
um);
                                                              newmouse(&MouseX,&MouseY,&press);
      }
                                                               re=pressed_anime(tra_d[0][0],tra_d[num/2][1],t
      else
                                                              ra_d[num-1][2],tra_d[num/2][3]);
      {
                                                                         if(re!=0)
                                                                         {
 tracktor_set_off0(xy[0],xy[1],x0-r,y0+r,2*r/num,
num);
                                                                              for(i=0;i<num;i++)
           clrmous(MouseX,MouseY);
                                                                              {
           draw_setoff(xy);
                                                                                  if(type[i]==0)
 start _ainime04_02(tra_d,num,des_x,des_y);
                                                                                        earth_cover01(x_p[i],
           for(i=0;i<num;i++)
                                                              y_p[i]);
           {
                                                                                        earth cover01(x p[i],
                                                              y_p[i] + 5);
 earth_fill03(x0-r+2*r/num+i*2*r/num,y0-r);
                                                                                  }
                                                                                  if(type[i]==1)
 earth_fill03(x0-r+2*r/num+i*2*r/num,y0-r-6);
           }
                                                               earth_cover02(x_p[i],y_p[i]);
 tracktor_return0(xy[0],xy[1],x0-r+2*r/num,y0-r,2
*r/num,num);
                                                               earth_cover02(x_p[i]+7,y_p[i]);
                                                                                  }
 picker_anime(xy[0],xy[1],des_x,des_y,2*r/num,n
                                                                              }
                                                                              if(mode==0)
um);
      }
                                                                              {
      return;
                                                                                   return;
    }
                                                                              }
                                                                              break;
    void start_ainime04_01(int (*tra_d)[4],int
                                                                         for(i=0;i<num;i++)
num,int *des x,int *des y)
    {
      int
                                                                              if(type[i]==4)
type[tracktor_num_max],i,x_p[tracktor_num_max
                                                                              {
                                                                                   count++;
      ,y_p[tracktor_num_max],cal_time[tracktor_
                                                                              }
num_max];
                                                                         for(i=0;i<num;i++)
      for(i=0;i<num;i++)
      {
                                                                         {
           type[i]=0;
                                                                              if(cal time[i]>=0)
```

```
cal_time[i]++;
                                                                    earth_fill03(x_p[i],y_p[i]-7);
                }
                                                                                             type[i]=1;
                                                                                             x_p[i]+=25;
 if(cal time[i]>0&&cal time[i]>=pick bar)
                                                                                        }
                     cal_time[i]=-1;
                                                                    delay(delaytime/(num-count));
                     if(type[i]==0)
                                                                                        continue;
                     {
                                                                                   }
                                                                                   else if(type[i]==1)
 draw_copak(x_p[i],y_p[i]+tracktor_l+co_pak_w);
                                                                                   {
                           des_x[i]=x_p[i];
                                                                                        earth_fill02(x_p[i],y_p[i]);
                                                                    init_tracktor01_b(x_p[i],y_p[i]);
 des_y[i]=y_p[i]+tracktor_l+co_pak_w;
                                                                                        y_p[i]++;
                      else
                                                                                        if(y_p[i]+40>=tra_d[i][3])
                     {
                                                                                             if(x_p[i]+25>tra_d[i][2])
 draw_copak(x_p[i],y_p[i]-2*co_pak_w);
                           des_x[i]=x_p[i];
                                                                                                   type[i]=3;
                                                                                             }
 des_y[i]=y_p[i]-2*co_pak_w;
                                                                                             else
                     }
                                                                                             {
                }
                                                                    earth_fill03(x_p[i],y_p[i]-1);
 if((type[i]!=0\&\&y_p[i]-40>=y_start+y_max)||x_p|
[i]>=tra_d[i][2])
                                                                    earth_fill03(x_p[i],y_p[i]+2);
                                                                                                   type[i]=2;
                                                                                                   x_p[i]+=25;
                     if(type[i]!=4)
                     {
                                                                                             }
                                                                                        }
 init_tracktor01_f(x_p[i]-25,y_p[i]);
                           type[i]=4;
                                                                    delay(delaytime/(num-count));
                     }
                                                                                        continue;
                                                                                   }
                     else
                     {
                                                                                   else
                           continue;
                                                                                   {
                     }
                                                                                        type[i]=2;
                                                                                        earth_fill01(x_p[i],y_p[i]);
                }
                else if(type[i]==0)
                {
                                                                    init_tracktor01_f(x_p[i],y_p[i]);
                      earth_fill01(x_p[i],y_p[i]);
                                                                                        y p[i]--;
                                                                                        if(y_p[i]<=tra_d[i][1])
 init_tracktor01_f(x_p[i],y_p[i]);
                                                                                        {
                     y_p[i]--;
                     if(y_p[i]<=tra_d[i][1])
                                                                    earth_fill03(x_p[i],y_p[i]-7);
                     {
                                                                                             type[i]=1;
```

```
x_p[i]+=25;
                                                               ],\
                    }
                                                                     y_p[tracktor_num_max],cal_time[tracktor_
                                                               num_max];
 delay(delaytime/(num-count));
                                                                     for(i=0;i<num;i++)
                    continue;
               }
                                                                          cal_time[i]=0;
               // else if(type[i]=3)
                                                                          type[i]=0;
               //{
                                                                          x_p[i]=tra_d[i][0];
               //
                    earth_fill02(x_p[i],y_p[i]);
                                                                          y_p[i]=y_start+y_max-40;
               //
                                                                     while(1)
 init_tracktor01_b(x_p[i],y_p[i]);
               // y_p[i]++;
                                                                     {
               //}
                                                                          int count=0,re=0;
           }
           for(i=0;i<num;i++)
                                                               newmouse(&MouseX,&MouseY,&press);
           {
                                                                re=pressed_anime(tra_d[0][0],tra_d[num/2-1][1
                draw_copak(des_x[i],des_y[i]);
                                                               ],tra_d[num-1][2],tra_d[num/2-1][3]);
                                                                          if(re!=0)
           if(count>=num)
           {
                                                                          {
                break;
                                                                               for(i=0;i<num;i++)
           }
      }
                                                                                    if(type[i]==0)
      for(i=0;i<num;i++)
                                                                                         earth_cover01(x_p[i],
           earth fill03(x p[i]-25,y p[i]);
                                                               y_p[i]);
           earth_fill03(x_p[i]-25,y_p[i]-6);
                                                                                         earth_cover01(x_p[i],
           if(cal_time[i]>=0)
                                                               y_p[i] + 5);
                                                                                    }
           {
                des_x[i]=0;
                                                                                    if(type[i]==1)
           }
                                                                                    {
           // else
           //{
                                                                earth_cover02(x_p[i],y_p[i]);
           //
               draw_copak(des_x[i],des_y[i]);
           //}
                                                                earth_cover02(x_p[i]+7,y_p[i]);
      }
                                                                                    }
      // if(cal_time[0]>=0)
      //{
                                                                               if(mode==0)
      //
           des_x[0]=0;
                                                                               {
      //}
                                                                                    return;
    }
                                                                               break;
    void start_ainime04_02(int (*tra_d)[4],int
num,int *des_x,int *des_y)
                                                                          for(i=0;i<num;i++)
    {
      int
                                                                               if(type[i]==4)
type[tracktor_num_max],i,x_p[tracktor_num_max
                                                                               {
```

```
earth_fill01(x_p[i],y_p[i]);
                     count++;
                }
           }
                                                                    init_tracktor02_f(x_p[i],y_p[i]);
           for(i=0;i<num;i++)
                                                                                        y_p[i]--;
                                                                                        if(y_p[i]<=tra_d[i][1])
           {
                if(cal_time[i]>=0)
                {
                                                                    earth_fill03(x_p[i],y_p[i]-7);
                     cal_time[i]++;
                                                                                              type[i]=1;
                                                                                              x p[i]+=25;
 if(cal_time[i]>0&&cal_time[i]>=pick_bar)
                                                                                        }
                                                                    delay(delaytime/(num-count));
                     cal_time[i]=-1;
                     if(type[i]==0)
                                                                                        continue;
                     {
                                                                                   }
                                                                                   else if(type[i]==1)
 draw_copak(x_p[i],y_p[i]+tracktor_l+co_pak_w);
                                                                                   {
                           des_x[i]=x_p[i];
                                                                                         earth_fill02(x_p[i],y_p[i]);
 des_y[i]=y_p[i]+tracktor_l+co_pak_w;
                                                                    init_tracktor02_b(x_p[i],y_p[i]);
                     }
                                                                                        y p[i]++;
                      else
                                                                                        if(y_p[i]+40>=tra_d[i][3])
                      {
                                                                                              if(x_p[i]+25>tra_d[i][2])
 draw_copak(x_p[i],y_p[i]-2*co_pak_w);
                                                                                              {
                           des x[i]=x p[i];
                                                                                                   type[i]=3;
                                                                                              }
 des_y[i]=y_p[i]-2*co_pak_w;
                                                                                              else
                     }
                                                                                              {
                }
                                                                    earth_fill03(x_p[i],y_p[i]-1);
 if((type[i]!=0\&\&y_p[i]-40>=y_start+y_max)||x_p|
[i]>=tra_d[i][2])
                                                                    earth_fill03(x_p[i],y_p[i]+2);
                                                                                                   type[i]=2;
                {
                                                                                                   x p[i] += 25;
                     if(type[i]!=4)
                     {
                                                                                             }
                                                                                        }
 init_tracktor02_f(x_p[i]-25,y_p[i]);
                           type[i]=4;
                                                                    delay(delaytime/(num-count));
                     }
                                                                                        continue;
                     else
                                                                                   }
                     {
                                                                                   else
                           continue;
                                                                                   {
                     }
                                                                                        type[i]=2;
                                                                                        earth_fill01(x_p[i],y_p[i]);
                }
                else if(type[i]==0)
                {
                                                                    init_tracktor02_f(x_p[i],y_p[i]);
```

```
y_p[i]--;
                                                if(y_p[i]<=tra_d[i][1])
                                                                                                                                                                     long int hellen(int x1, int y1, int x2, int y2, int
                                                                                                                                                         x3, int y3)
                                                                                                                                                                     {
earth_fill03(x_p[i],y_p[i]-7);
                                                                                                                                                                         long int a = sqrt((x1 - x2) * (x1 - x2) + (y1 -
                                                                                                                                                         y2) * (y1 - y2)), b = sqrt((x1 - x3) * (x1 - x3) + (y1 - x3))
                                                            type[i]=1;
                                                            x p[i] += 25;
                                                                                                                                                         y3) * (y1 - y3)),
                                                }
                                                                                                                                                                                                    c = sqrt((x3 - x2) * (x3 - x2) + (y3)
                                                                                                                                                         -y2)*(y3-y2)), p = (a + b + c) / 2;
delay(delaytime/(num-count));
                                                continue;
                                                                                                                                                                         return sqrt(p * (p - a) * (p - b) * (p - c));
                                                                                                                                                                     }
                                   // else if(type[i]=3)
                                   //{
                                                                                                                                                                     long int cal_poly_field(int *x, int *y, int *flag)
                                   // earth_fill02(x_p[i],y_p[i]);
                                   //
                                                                                                                                                                         int x0 = (x[0] + x[(*flag) / 2]) / 2, y0 = (y[0] + y[0])
init_tracktor02_b(x_p[i],y_p[i]);
                                                                                                                                                         y[(*flag) / 2]) / 2, i;
                                                                                                                                                                         long int s_field = 0;
                                   // y_p[i]++;
                                   //}
                                                                                                                                                                        for (i = 0; i < (*flag) - 2; i++)
                        }
                        for(i=0;i<num;i++)
                                                                                                                                                                                     s field += hellen(x0, y0, x[i], y[i], x[i +
                                                                                                                                                         1], y[i + 1]);
                        {
                                    draw_copak(des_x[i],des_y[i]);
                                                                                                                                                                        }
                        }
                                                                                                                                                                         s_{i} = hellen(x_{i}, x_{i}, x_{i},
                        if(count>=num)
                                                                                                                                                                         return s_field;
                        {
                                                                                                                                                                     }
                                    break;
                                                                                                                                                                      double cal_circle_field(int r)
           }
                                                                                                                                                                         double pi = 3.1415926;
           for(i=0;i<num;i++)
                                                                                                                                                                         return pi * r * r;
           {
                        earth_fill03(x_p[i]-25,y_p[i]);
                                                                                                                                                                     }
                        earth_fill03(x_p[i]-25,y_p[i]-6);
                        if(cal_time[i]>=0)
                                                                                                                                                                     void select setoff01(int *xy,int x end,int
                        {
                                    des_x[i]=0;
                                                                                                                                                         y_end)
                        }
                                                                                                                                                                     {
                        // else
                                                                                                                                                                         //int i=0;
                       //{
                                                                                                                                                                         cleardevice();
                                                                                                                                                                         setbkcolor(WHITE);
                                   draw_copak(des_x[i],des_y[i]);
                        //}
                                                                                                                                                                         puthz(150,30,"请在框外选择农机出发点
           }
                                                                                                                                                         ",32,32,BLUE);
                                                                                                                                                                         init based field();
           // if(cal_time[0]>=0)
           //{
                                                                                                                                                                         setlinestyle(0,0,3);
           // des_x[0]=0;
                                                                                                                                                                         setcolor(GREEN);
           //}
                                                                                                                                                                         rectangle(x_start,y_start,x_end,y_end);
        }
                                                                                                                                                                         // for(i=0;i<100;i++)
```

```
//{
                                                                                                                                                                                                                                                                                               setcolor(GREEN);
                         //
                                                                                                                                                                                                                                                                                               rectangle(x_s,y_s,x_e,y_e);
      newmouse(&MouseX,&MouseY,&press);
                                                                                                                                                                                                                                                                                              // for(i=0;i<100;i++)
                         // delay(10);
                                                                                                                                                                                                                                                                                              //{
                         //}
                                                                                                                                                                                                                                                                                               //
                         while(1)
                                                                                                                                                                                                                                                                           newmouse(&MouseX,&MouseY,&press);
                           {
                                                                                                                                                                                                                                                                                               // delay(10);
                                                                                                                                                                                                                                                                                              //}
      newmouse(&MouseX,&MouseY,&press);
                                                                                                                                                                                                                                                                                               while(1)
                                                                                                                                                                                                                                                                                               {
      if(mouse_press(0,y_start,x_start,480)==1||mous
e_press(x_start,y_end,x_end,480)==1||mouse_pr
                                                                                                                                                                                                                                                                           newmouse(&MouseX,&MouseY,&press);
ess(x_end,y_start,615,480)==1)
                                             {
                                                                                                                                                                                                                                                                           if(mouse_press(0,y_start,x_s,480)==1||mouse_
                                                                  xy[0]=MouseX;
                                                                                                                                                                                                                                                                      press(x_s,y_e,x_e,480)==1\
                                                                  xy[1]=MouseY;
                                                                  break;
                                                                                                                                                                                                                                                                           ||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_press(x_e,y_start,615,480)==1||mouse_p
                                                                                                                                                                                                                                                                     e_press(x_s,y_start,x_e,y_s)==1)
                                              delay(20);
                         }
                                                                                                                                                                                                                                                                                                                                       xy[0]=MouseX;
                          if(xy[0] \le x \text{ start} = x \text{ start } = x \text{ star
                                                                                                                                                                                                                                                                                                                                       xy[1]=MouseY;
rt)
                                                                                                                                                                                                                                                                                                                                       break;
                         {
                                                                                                                                                                                                                                                                                                                  }
                                                                                                                                                                                                                                                                                                                   delay(20);
                                              xy[0]=x_start-tra_start_l;
                         }
                           if(xy[0]>=x start&&xy[0]<=x end&&xy[1]<=
                                                                                                                                                                                                                                                                                              // while(1)
y_end)
                                                                                                                                                                                                                                                                                              //{
                                                                                                                                                                                                                                                                                                                  if(bioskey(0)==p_Enter)
                         {
                                              xy[1]=y_end;
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                  {
                         }
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                                                       break;
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                  }
                          clrmous(MouseX,MouseY);
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                 if(flag==0)
                          draw_setoff(xy);
                                                                                                                                                                                                                                                                                               //
                         choose_speed();
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                                                       itoa(100*times,str,5);
                           return;
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                                                       outtextxy(450,40,str);
                    }
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                                       flag=1;
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                  }
                    void select_setoff02(int *xy,int x_s,int y_s,int
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                                  if(times>0.5&&
x_e,int y_e)
                                                                                                                                                                                                                                                                     bioskey(0)==p_Up_arrow)
                    {
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                               {
                          cleardevice();
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                                       times-=0.05;
                          setbkcolor(WHITE);
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                                       flag=0;
                           puthz(100,30,"请在框外选择农机出发点
                                                                                                                                                                                                                                                                                              //
                                                                                                                                                                                                                                                                                                              }
",32,32,BLUE);
                         //puthz(400,40,"速度",32,32,GREEN);
                                                                                                                                                                                                                                                                           if(times<2&&bioskey(0)==p_Donw_arrow)
                          init based field();
                                                                                                                                                                                                                                                                                              // {
                          setlinestyle(0,0,3);
                                                                                                                                                                                                                                                                                               //
                                                                                                                                                                                                                                                                                                                                       times+=0.05;
```

```
//
                                                                          }
                flag=0;
                                                                          if
      //
           delay(delaytime);
                                                               (mouse_press(320-100,85,320+100,95) == 2)
      //}
                                                                          {
      if(xy[0] \le x_s \& xy[0] + tra_start_l \ge x_s)
                                                                               MouseS = 1;
           xy[0]=x_s-tra_start_l;
                                                                          delay(delaytime);
      }
      if(xy[0]+tra_start_l>=615)
                                                                     if(x_p \le 20)
           xy[0]=615-tra_start_l;
                                                                          x_p=20;
      if(xy[0]>=x_s\&xy[0]<=x_e\&xy[1]<=y_e)
                                                                     x_p/=200;
                                                                     delaytime*=x_p;
                                                                     clrmous(MouseX,MouseY);
           xy[1]=y_e;
      }
                                                                    }
      if(xy[1]<y_s+tra_start_d)
                                                                    void draw_setoff(int *xy)
           xy[1]=y_s+tra_start_d;
      }
                                                                     setfillstyle(1,LIGHTGRAY);
      clrmous(MouseX,MouseY);
                                                                     bar(xy[0],xy[1],xy[0]+tra_start_l,xy[1]+tra_s
      draw setoff(xy);
                                                               tart_d);
      choose_speed();
                                                                    }
      return;
    }
    void choose_speed()
                                                                    void
                                                                              tracktor_set_off(int
                                                                                                       start x,int
                                                               start_y,int des_x,int des_y,int distance,int num)
      float x_p=0;
      setcolor(RED);
                                                                     if (start_x < des_x && start_y < des_y)
      setfillstyle(1,WHITE);
                                                                                                          start_y,
      bar(0,0,480,70);
                                                                          tracktor_set_off01(start_x,
      puthz(100,30,"请在横轴上选择农机速度
                                                               des_x, des_y, distance, num);
值",32,32,BLUE);
      setfillstyle(1,RED);
                                                                     else if (start x < des x && start y > des y)
      bar(320-100,85,320,95);
                                                                     {
      setfillstyle(1,GREEN);
                                                                          tracktor_set_off02(start_x,
                                                                                                          start_y,
      bar(320,85,320+100,95);
                                                               des_x, des_y, distance, num);
      while(1)
                                                                     }
      {
                                                                     else if (start_x > des_x && start_y < des_y)
                                                                          tracktor set off03(start x,
 newmouse(&MouseX,&MouseY,&press);
                                                                                                          start y,
                                                               des_x, des_y, distance, num);
 if(mouse_press(320-100,85,320+100,95)==1)
                                                                     }
                                                                     else
                x p=420-MouseX;
                break;
                                                                          tracktor_set_off04(start_x,
                                                                                                          start y,
```

```
des_x, des_y, distance, num);
                                                                                   }
                                                                                   if(mode==0)
     }
                                                                                   {
                                                                                       return;
     // start x<des x&&start y<des y
     void tracktor_set_off01(int start_x, int start_y,
                                                                                   break;
int des_x, int des_y, int distance, int num)
                                                                             for (i = 0; i < num; i++)
     {
      int
                          x_p[tracktor_num_max],
                                                                                   if (time[i] > 0)
y_p[tracktor_num_max],
type[tracktor_num_max],
                                                                                   {
time[tracktor_num_max], i;
                                                                                       time[i]--;
      for (i = 0; i < num; i++)
                                                                                       continue;
                                                                                   }
           type[i] = 0;
                                                                                   if (type[i] == 2)
           time[i] = i * distance;
                                                                                   {
           x_p[i] = (2 * start_x + tra_start_l) / 2 -
                                                                                        count++;
tracktor_w / 2;
                                                                                        continue;
                                                                                   }
           y_p[i] = start_y + tra_start_d + 1;
      }
                                                                                   if (type[i] == 0)
      //clrmous(MouseX,MouseY);
                                                                                   {
      while (1)
                                                                                        earth_cover01(x_p[i], y_p[i]);
      {
                                                                                       y_p[i]++;
           int count = 0,re=0;
                                                                                        init_tracktor01_b(x_p[i],
                                                                  y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                                        if (y_p[i] >= des_y)
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                                             earth_cover01(x_p[i],
);
                                                                  y_p[i]);
           if(re!=0)
                                                                                             earth_cover01(x_p[i],
           {
                                                                  y_p[i] + 5);
                for(i=0;i<num;i++)
                                                                                             type[i] = 1;
                                                                                             x_p[i] += tracktor_l;
                     if(type[i]==0)
                                                                                             init_tracktor01_r(x_p[i],
                     {
                                                                  y_p[i]);
                                                                                             continue;
                           earth_cover01(x_p[i],
                                                                                       }
y_p[i]);
                           earth_cover01(x_p[i],
                                                                                   }
y_p[i] + 5);
                                                                                   if (type[i] == 1)
                     }
                                                                                   {
                     if(type[i]==1)
                                                                                        earth_cover02(x_p[i], y_p[i]);
                                                                                        x p[i]++;
                                                                                        init_tracktor01_r(x_p[i],
 earth_cover02(x_p[i],y_p[i]);
                                                                  y_p[i]);
                                                                                        if (x_p[i] >= des_x + (num - i -
 earth_cover02(x_p[i]+7,y_p[i]);
                                                                  1) * distance-tracktor I)
                     }
                                                                                       {
```

```
for(i=0;i<num;i++)
 earth_cover02(x_p[i],y_p[i]);
                                                                                  {
                                                                                       if(type[i]==0)
 earth_cover02(x_p[i]+7,y_p[i]);
                                                                                       {
                          type[i]=2;
                                                                                            earth cover01(x p[i],
                                                                 y_p[i]);
 //init_tracktor01_f(x_p[i],y_p[i]-35);
                                                                                            earth_cover01(x_p[i],
                          continue;
                                                                 y_p[i] -7);
                     }
                                                                                       }
                }
                                                                                       if(type[i]==1)
           }
                                                                                       {
           if (count >= num)
                                                                   earth_cover02(x_p[i],y_p[i]);
           {
                break;
                                                                   earth_cover02(x_p[i]+7,y_p[i]);
           delay(delaytime);
      }
                                                                                  }
     }
                                                                                  if(mode==0)
     // start_x<des_x&&start_y>des_y
                                                                                       return;
     void tracktor set off02(int start x, int start y,
                                                                                  }
int des_x, int des_y, int distance, int num)
                                                                                  break;
     {
                                                                             }
                          x_p[tracktor_num_max],
                                                                             for (i = 0; i < num; i++)
      int
y_p[tracktor_num_max],
type[tracktor num max],
                                                                                  if (time[i] > 0)
time[tracktor_num_max], i;
                                                                                  {
      for (i = 0; i < num; i++)
                                                                                       time[i]--;
      {
                                                                                       continue;
           type[i] = 0;
                                                                                  }
           time[i] = i * distance;
                                                                                  if (type[i] == 2)
           x_p[i] = (2 * start_x + tra_start_l) / 2 -
tracktor_w / 2;
                                                                                       count++;
           y_p[i] = start_y - tra_start_d - 1 -
                                                                                       continue;
tracktor I;
                                                                                  }
                                                                                  if (type[i] == 0)
      }
      //clrmous(MouseX,MouseY);
                                                                                  {
      while (1)
                                                                                       earth_cover01(x_p[i], y_p[i]);
      {
                                                                                       y_p[i]--;
           int count = 0,re=0;
                                                                                       init_tracktor01_f(x_p[i],
                                                                 y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                                       if (y p[i] \le des y)
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                                            earth_cover01(x_p[i],
);
                                                                 y_p[i]);
           if(re!=0)
                                                                                            earth_cover01(x_p[i],
           {
                                                                 y_p[i] - 7);
```

```
type[i] = 1;
                                                                             x_p[i] = (2 * start_x + tra_start_l) / 2 -
                           x_p[i] += tracktor_l;
                                                                  tracktor_w / 2;
                           init_tracktor01_r(x_p[i],
                                                                             y_p[i] = start_y + tra_start_d + 1;
y_p[i]);
                           continue;
                                                                        //clrmous(MouseX,MouseY);
                     }
                                                                        while (1)
                }
                                                                        {
                if (type[i] == 1)
                                                                             int count = 0,re=0;
                     earth_cover02(x_p[i], y_p[i]);
                                                                   newmouse(&MouseX,&MouseY,&press);
                     x_p[i]++;
                     init_tracktor01_r(x_p[i],
                                                                   re=pressed_anime(x_start,y_start,x_start,y_start
y_p[i]);
                                                                 );
                     if (x p[i] >= des x + (num - i -
                                                                             if(re!=0)
1) * distance-tracktor_l)
                                                                             {
                     {
                                                                                  for(i=0;i<num;i++)
                                                                                  {
 earth_cover02(x_p[i],y_p[i]);
                                                                                       if(type[i]==0)
                                                                                       {
 earth_cover02(x_p[i]+7,y_p[i]);
                                                                                             earth_cover01(x_p[i],
                           type[i]=2;
                                                                 y_p[i]);
                                                                                            earth_cover01(x_p[i],
 //init_tracktor01_f(x_p[i],y_p[i]-35);
                                                                 y_p[i] + 5);
                                                                                       }
                           continue;
                                                                                       if(type[i]==1)
                     }
                }
           }
           if (count >= num)
                                                                   earth_cover02(x_p[i]+2,y_p[i]);
           {
                break;
                                                                   earth_cover02(x_p[i]-5,y_p[i]);
                                                                                       }
      delay(delaytime);
                                                                                  }
      }
                                                                                  if(mode==0)
     }
                                                                                  {
                                                                                       return;
     // start_x>des_x&&start_y<des_y
                                                                                  }
     void tracktor_set_off03(int start_x, int start_y,
                                                                                  break;
int des_x, int des_y, int distance, int num)
     {
                                                                             for (i = 0; i < num; i++)
      int
                          x_p[tracktor_num_max],
                                                                             {
                                                                                  if (time[i] > 0)
y_p[tracktor_num_max],
type[tracktor num max],
                                                                                       time[i]--;
time[tracktor_num_max], i;
      for (i = 0; i < num; i++)
                                                                                       continue;
      {
                                                                                  }
           type[i] = 0;
                                                                                  if (type[i] == 2)
           time[i] = i * distance;
                                                                                  {
```

```
delay(delaytime);
                     count++;
                                                                       }
                     continue;
                }
                                                                      }
                if (type[i] == 0)
                                                                      // start x>=des x&&start y>=des y
                     earth_cover01(x_p[i], y_p[i]);
                                                                      void tracktor_set_off04(int start_x, int start_y,
                                                                 int des_x, int des_y, int distance, int num)
                     y_p[i]++;
                     init_tracktor01_b(x_p[i],
                                                                      {
y_p[i]);
                                                                       int
                                                                                           x_p[tracktor_num_max],
                                                                 y_p[tracktor_num_max],
                     if (y p[i] >= des y)
                                                                 type[tracktor_num_max],
                          earth_cover01(x_p[i],
                                                                 time[tracktor num max], i;
y_p[i]);
                                                                       for (i = 0; i < num; i++)
                          earth_cover01(x_p[i],
y_p[i] + 5);
                                                                            type[i] = 0;
                          type[i] = 1;
                                                                            time[i] = i * distance;
                          x_p[i] += tracktor_l;
                                                                            x_p[i] = (2 * start_x + tra_start_l) / 2 -
                          init_tracktor01_l(x_p[i],
                                                                 tracktor_w / 2;
y_p[i]);
                                                                            y_p[i] = start_y - tra_start_d - 1 -
                                                                 tracktor_l;
                          continue;
                     }
                }
                                                                       //clrmous(MouseX,MouseY);
                if (type[i] == 1)
                                                                       while (1)
                     earth\_cover02(x\_p[i] + 5,
                                                                            int count = 0,re=0;
y p[i]);
                     x_p[i]--;
                                                                  newmouse(&MouseX,&MouseY,&press);
                     init_tracktor01_l(x_p[i],
y_p[i]);
                                                                  re=pressed_anime(x_start,y_start,x_start,y_start
                     if (x_p[i] \le des_x + i *
                                                                 );
distance+tracktor I)
                                                                            if(re!=0)
                                                                            {
                                                                                 for(i=0;i<num;i++)
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                                 {
                                                                                      if(type[i]==0)
 earth_cover02(x_p[i]-5,y_p[i]);
                                                                                      {
                          type[i]=2;
                                                                                            earth_cover01(x_p[i],
                                                                 y_p[i]);
 //init_tracktor01_f(x_p[i]-50,y_p[i]-35);
                                                                                            earth_cover01(x_p[i],
                          continue;
                                                                 y_p[i] -7);
                     }
                }
                                                                                      if(type[i]==1)
           }
                                                                                      {
           if (count >= num)
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
                break;
           }
                                                                  earth_cover02(x_p[i]-5,y_p[i]);
```

```
}
                                                                 distance+tracktor_I)
                if(mode==0)
                                                                   earth_cover02(x_p[i]+2,y_p[i]);
                      return;
                                                                   earth_cover02(x_p[i]-5,y_p[i]);
                break;
                                                                                            type[i]=2;
           }
           for (i = 0; i < num; i++)
                                                                   //init_tracktor01_f(x_p[i]-50,y_p[i]-35);
                                                                                            continue;
           {
                if (time[i] > 0)
                                                                                       }
                                                                                  }
                     time[i]--;
                                                                             }
                     continue;
                                                                             if (count >= num)
                if (type[i] == 2)
                                                                                  break;
                                                                             delay(delaytime);
                     count++;
                                                                        }
                     continue;
                                                                      }
                }
                if (type[i] == 0)
                                                                      void tracktor_return(int start_x, int start_y,
                     earth_cover01(x_p[i], y_p[i]);
                                                                 int des_x, int des_y, int distance, int num)
                                                                      {
                     y_p[i]--;
                     init_tracktor01_f(x_p[i],
                                                                        if(start_x<des_x&&start_y>des_y)
y_p[i]);
                                                                        {
                     if (y_p[i] \le des_y)
                                                                             tracktor_return01(start_x,
                                                                                                              start_y,
                                                                 des_x, des_y, distance, num);
                          earth_cover01(x_p[i],
                                                                        }
y_p[i]);
                                                                        else if(start_x>des_x&&start_y>des_y)
                          earth_cover01(x_p[i],
                                                                        {
y_p[i] - 7);
                                                                   tracktor_return02(start_x,start_y,des_x,des_y,di
                          type[i] = 1;
                          x_p[i] += tracktor_l;
                                                                 stance, num);
                          init_tracktor01_r(x_p[i],
y_p[i]);
                                                                        else if(start_x<des_x&&start_y<des_y)
                          continue;
                                                                        {
                     }
                }
                                                                   tracktor_return03(start_x,start_y,des_x,des_y,di
                if (type[i] == 1)
                                                                 stance, num);
                                                                        }
                     earth cover02(x p[i] + 7,
                                                                        else
y_p[i]);
                                                                        {
                     x_p[i]--;
                     init_tracktor01_l(x_p[i],
                                                                   tracktor_return04(start_x,start_y,des_x,des_y,di
y_p[i]);
                                                                 stance,num);
                     if (x p[i] \le des x + i *
                                                                        }
```

```
}
                                                                                 }
                                                                                 if(mode==0)
     void
              tracktor return01(int
                                         start x,int
                                                                                 {
start_y,int des_x,int des_y,int distance,int num)
                                                                                      return;
      int
                                                                                 break;
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                            }
                                                                            for(i=0;i<num;i++)
type[tracktor_num_max],i;
      if(distance<tracktor_l)
      {
                                                                                 if(type[i]==2)
           distance=tracktor_l;
                                                                                 {
      }
                                                                                      count++;
      for(i=0;i<num;i++)
                                                                                      continue;
      {
                                                                                 }
           type[i]=0;
                                                                                 if(type[i]==0)
           x_p[i]=des_x+i*distance;
                                                                                 {
           y_p[i]=des_y;
           earth_fill03(x_p[i],y_p[i]);
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
           x_p[i]-=tracktor_l;
                                                                                      x_p[i]--;
           y_p[i]-=tracktor_w;
      }
                                                                  init tracktor01 l(x p[i],y p[i]);
      while(1)
                                                                  if(x_p[i] \le ((2*start_x+tra_start_l)/2-tracktor_w/
      {
           int count = 0,re=0;
                                                                 2)+50)
                                                                                      {
 newmouse(&MouseX,&MouseY,&press);
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
);
                                                                  earth_cover02(x_p[i]-6,y_p[i]);
           if(re!=0)
                                                                  x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
           {
                for(i=0;i<num;i++)
                                                                                           type[i]=1;
                                                                                           continue;
                     if(type[i]==0)
                                                                                      }
                                                                                 }
                                                                                 if(type[i]==1)
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                                 {
                                                                                      earth_cover01(x_p[i],y_p[i]);
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                                      y_p[i]++;
                     }
                     if(type[i]==1)
                                                                  init_tracktor01_b(x_p[i],y_p[i]);
                                                                                      if(y p[i]>=start y-48)
                                                                                      {
 earth_cover01(x_p[i],y_p[i]);
                                                                  earth_cover01(x_p[i],y_p[i]);
 earth_cover01(x_p[i],y_p[i]+5);
                     }
                                                                  earth_cover01(x_p[i],y_p[i]+5);
```

```
type[i]=2;
                                                                 earth_cover02(x_p[i]+6,y_p[i]);
                          continue;
                     }
                                                                 earth_cover02(x_p[i]-2,y_p[i]);
                }
           }
                                                                                     if(type[i]==1)
           if(count>=num)
           {
                break;
                                                                 earth_cover01(x_p[i],y_p[i]);
           }
           delay(delaytime);
                                                                 earth_cover01(x_p[i],y_p[i]+5);
      }
                                                                                     }
    }
                                                                                }
                                                                                if(mode==0)
     void
              tracktor_return02(int
                                        start x,int
                                                                                {
start_y,int des_x,int des_y,int distance,int num)
                                                                                     return;
     {
                                                                                }
                                                                                break;
      int
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                           }
type[tracktor_num_max],i;
                                                                           for(i=0;i<num;i++)
      if(distance<tracktor_l)
                                                                           {
      {
                                                                                if(type[i]==2)
           distance=tracktor_l;
                                                                                {
      }
                                                                                     count++;
      for(i=0;i<num;i++)
                                                                                     continue;
                                                                                }
                                                                                if(type[i]==0)
           type[num-1-i]=0;
           x_p[num-1-i]=des_x+i*distance;
                                                                                {
           y_p[num-1-i]=des_y;
           earth_fill03(x_p[i],y_p[i]);
                                                                 earth_cover02(x_p[i]+2,y_p[i]);
           x_p[num-1-i]+=tracktor_l;
                                                                                     x_p[i]++;
           y_p[num-1-i]-=tracktor_w;
      }
                                                                 init_tracktor01_r(x_p[i],y_p[i]);
      while(1)
      {
                                                                 if(x_p[i]>=((2*start_x+tra_start_l)/2-tracktor_w/
           int count = 0,re=0;
                                                                2))
                                                                                     {
 newmouse(&MouseX,&MouseY,&press);
                                                                 earth_cover02(x_p[i]+6,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
);
                                                                 earth_cover02(x_p[i]-2,y_p[i]);
           if(re!=0)
                                                                 x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
                for(i=0;i<num;i++)
                                                                                          type[i]=1;
                                                                                          continue;
                                                                                     }
                     if(type[i]==0)
                                                                                }
                                                                                if(type[i]==1)
```

```
{
                                                                  newmouse(&MouseX,&MouseY,&press);
                     earth_cover01(x_p[i],y_p[i]);
                     y_p[i]++;
                                                                  re=pressed_anime(x_start,y_start,x_start,y_start
                                                                 );
 init_tracktor01_b(x_p[i],y_p[i]);
                                                                            if(re!=0)
                     if(y_p[i] > = start_y-48)
                                                                            {
                                                                                 for(i=0;i<num;i++)
                     {
                                                                                 {
 earth_cover01(x_p[i],y_p[i]);
                                                                                       if(type[i]==0)
 earth_cover01(x_p[i],y_p[i]+5);
                          type[i]=2;
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
                          continue;
                     }
                                                                  earth_cover02(x_p[i]-6,y_p[i]);
                }
           }
                                                                                      if(type[i]==1)
           if(count>=num)
                                                                                      {
           {
                break;
                                                                  earth_cover01(x_p[i],y_p[i]-7);
           }
           delay(delaytime);
                                                                  earth_cover01(x_p[i],y_p[i]+3);
      }
                                                                                      }
    }
                                                                                 }
                                                                                 if(mode==0)
              tracktor_return03(int
                                         start_x,int
                                                                                 {
start y,int des x,int des y,int distance,int num)
                                                                                       return;
     {
                                                                                 }
      int
                                                                                 break;
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                            }
type[tracktor_num_max],i;
                                                                            for(i=0;i<num;i++)
      if(distance<tracktor_l)</pre>
                                                                            {
      {
                                                                                 if(type[i]==2)
           distance=tracktor_l;
                                                                                 {
      }
                                                                                       count++;
      for(i=0;i<num;i++)
                                                                                      continue;
      {
                                                                                 }
           type[i]=0;
                                                                                 if(type[i]==0)
           x_p[i]=des_x+i*distance;
                                                                                 {
           y_p[i]=des_y;
           earth_fill03(x_p[i],y_p[i]);
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
           x_p[i]-=tracktor_l;
                                                                                      x_p[i]--;
           y p[i]-=tracktor w;
      }
                                                                  init_tracktor01_l(x_p[i],y_p[i]);
      while(1)
                                                                  if(x_p[i]<=((2*start_x+tra_start_I)/2-tracktor_w/
           int count = 0,re=0;
                                                                 2)+50)
                                                                                      {
```

```
for(i=0;i<num;i++)
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                            type[num-1-i]=0;
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                  x p[num-1-i]=des x+i*distance+tracktor l;
 x_p[i]=(2*start_x+tra_start_l)/2-tracktor_w/2;
                                                                            y_p[num-1-i]=des_y-tracktor_w;
                                                                            earth_fill03(x_p[i],y_p[i]);
                          y_p[i]-=tracktor_l;
                          type[i]=1;
                                                                       }
                                                                       while(1)
                          continue;
                     }
                                                                       {
                                                                            int count = 0,re=0;
                }
                if(type[i]==1)
                                                                  newmouse(&MouseX,&MouseY,&press);
                {
                     earth_cover01(x_p[i],y_p[i]);
                     y_p[i]--;
                                                                  re=pressed_anime(x_start,y_start,x_start,y_start
                                                                 );
 init_tracktor01_f(x_p[i],y_p[i]);
                                                                            if(re!=0)
                                                                            {
 if(y_p[i]<=start_y+tra_start_l-6)</pre>
                                                                                 for(i=0;i<num;i++)
                                                                                 {
                                                                                      if(type[i]==0)
 earth_cover01(x_p[i],y_p[i]-7);
 earth_cover01(x_p[i],y_p[i]+3);
                                                                  earth_cover02(x_p[i]+6,y_p[i]);
                          type[i]=2;
                          continue;
                                                                  earth_cover02(x_p[i]-2,y_p[i]);
                     }
                }
                                                                                      if(type[i]==1)
           }
           if(count>=num)
                                                                  earth_cover01(x_p[i],y_p[i]-7);
           {
                break;
                                                                  earth_cover01(x_p[i],y_p[i]+3);
           delay(delaytime);
                                                                                      }
      }
                                                                                 }
    }
                                                                                 if(mode==0)
                                                                                 {
              tracktor_return04(int
                                        start x,int
                                                                                      return;
start_y,int des_x,int des_y,int distance,int num)
                                                                                 }
     {
                                                                                 break;
x p[tracktor num max],y p[tracktor num max],
                                                                            for(i=0;i<num;i++)
type[tracktor_num_max],i;
                                                                            {
      if(distance<tracktor_l)</pre>
                                                                                 if(type[i]==2)
      {
                                                                                 {
           distance=tracktor_l;
                                                                                      count++;
      }
                                                                                      continue;
```

```
}
                if(type[i]==0)
                                                                      void
                                                                               tracktor set offO(int
                                                                                                          start x,int
                                                                 start_y,int des_x,int des_y,int distance,int num)
 earth_cover02(x_p[i]+2,y_p[i]);
                     x_p[i]++;
                                                                       if(start_x<des_x&&start_y<des_y)
 init_tracktor01_r(x_p[i],y_p[i]);
                                                                  tracktor_set_off001(start_x,start_y,des_x,des_y,
 if(x p[i]>=((2*start x+tra start I)/2-tracktor w/
                                                                 distance, num);
2))
                     {
                                                                       else if(start x<des x&&start y>des y)
                                                                       {
 earth_cover02(x_p[i]+6,y_p[i]);
                                                                  tracktor_set_off002(start_x,start_y,des_x,des_y,
 earth_cover02(x_p[i]-2,y_p[i]);
                                                                 distance, num);
 x_p[i]=(2*start_x+tra_start_l)/2-tracktor_w/2;
                                                                       else if(start_x>des_x&&start_y<des_y)
                          y_p[i]-=tracktor_w;
                          type[i]=1;
                          continue;
                                                                  tracktor set off003(start x,start y,des x,des y,
                     }
                                                                 distance, num);
                                                                       }
                if(type[i]==1)
                                                                       else
                                                                       {
                     earth cover01(x p[i],y p[i]);
                     y_p[i]--;
                                                                  tracktor_set_off004(start_x,start_y,des_x,des_y,
                                                                 distance, num);
 init_tracktor01_f(x_p[i],y_p[i]);
                                                                       }
                                                                      }
 if(y_p[i]<=start_y+tra_start_l-6)
                                                                      //start_x<des_x&&start_y<des_y
                                                                              tracktor_set_off001(int
                                                                                                          start x,int
 earth_cover01(x_p[i],y_p[i]-7);
                                                                 start_y,int des_x,int des_y,int distance,int num)
 earth_cover01(x_p[i],y_p[i]+3);
                                                                       int
                                                                                           x p[tracktor num max],
                          type[i]=2;
                                                                 y_p[tracktor_num_max],
                          continue;
                                                                 type[tracktor num max],
                     }
                                                                 time[tracktor_num_max], i;
                }
                                                                       for (i = 0; i < num; i++)
           if(count>=num)
                                                                            type[i] = 0;
                                                                            time[i] = i * distance;
           {
                break;
                                                                            x_p[i] = (2 * start_x + tra_start_l) / 2 -
                                                                 tracktor_w / 2;
           delay(delaytime);
                                                                            y_p[i] = start_y + tra_start_d + 1;
      }
                                                                       }
```

```
while (1)
                                                                                       earth_cover01(x_p[i], y_p[i]);
      {
                                                                                       y_p[i]++;
           int count = 0,re=0;
                                                                   init_tracktor02_b(x_p[i],y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                                       if(y_p[i] > = des_y)
 re=pressed_anime(x_start,y_start,x_start,y_start
);
                                                                   earth_cover01(x_p[i],y_p[i]);
           if(re!=0)
           {
                                                                   earth_cover01(x_p[i],y_p[i]+5);
                for(i=0;i<num;i++)
                                                                                            type[i]=1;
                                                                                            x_p[i]+=tracktor_l;
                     if(type[i]==0)
                     {
                                                                   init_tracktor02_r(x_p[i],y_p[i]);
                                                                                            continue;
 earth_cover01(x_p[i],y_p[i]);
                                                                                       }
                                                                                  }
 earth_cover01(x_p[i],y_p[i]+5);
                                                                                  if (type[i] == 1)
                                                                                       earth_cover02(x_p[i], y_p[i]);
                     if(type[i]==1)
                                                                                       x p[i]++;
 earth_cover02(x_p[i],y_p[i]);
                                                                   init_tracktor02_r(x_p[i],y_p[i]);
                                                                   if(x_p[i]>=des_x+(num-i-1)*distance-tracktor_l)
 earth_cover02(x_p[i]+7,y_p[i]);
                if(mode==0)
                                                                   earth_cover02(x_p[i],y_p[i]);
                                                                   earth_cover02(x_p[i]+7,y_p[i]);
                     return;
                                                                                            type[i]=2;
                break;
                                                                   //init_tracktor02_f(x_p[i],y_p[i]-35);
           }
           for (i = 0; i < num; i++)
                                                                                            continue;
                                                                                       }
                if (time[i] > 0)
                                                                                  }
                {
                                                                             }
                     time[i]--;
                                                                             if(count>=num)
                     continue;
                                                                                  break;
                if (type[i] == 2)
                                                                             delay(delaytime);
                                                                        }
                     count++;
                     continue;
                                                                       }
                if (type[i] == 0)
                                                                       //start_x<des_x&&start_y>des_y
                {
                                                                       void
                                                                               tracktor_set_off002(int
                                                                                                           start x,int
```

```
for(i=0;i<num;i++)
start_y,int des_x,int des_y,int distance,int num)
      int
                                                                                 if(time[i]>0)
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                                 {
type[tracktor num max],time[tracktor num max
                                                                                      time[i]--;
],i;
                                                                                      continue;
      for(i=0;i<num;i++)
                                                                                 }
                                                                                 if(type[i]==2)
      {
           type[i]=0;
           time[i]=i*distance;
                                                                                      count++;
                                                                                      continue;
 x_p[i]=(2*start_x+tra_start_l)/2-tracktor_w/2;
                                                                                 }
                                                                                 if(type[i]==0)
           y_p[i]=start_y-tra_start_d-1-tracktor_l;
      }
                                                                                 {
      while(1)
                                                                                      earth_cover01(x_p[i],y_p[i]);
      {
                                                                                      y_p[i]--;
           int count = 0,re=0;
                                                                  init_tracktor02_f(x_p[i],y_p[i]);
                                                                                      if(y_p[i]<=des_y)
 newmouse(&MouseX,&MouseY,&press);
                                                                                      {
 re=pressed anime(x start,y start,x start,y start
);
                                                                  earth_cover01(x_p[i],y_p[i]);
           if(re!=0)
           {
                                                                  earth_cover01(x_p[i],y_p[i]-7);
                for(i=0;i<num;i++)
                                                                                           type[i]=1;
                {
                                                                                           x p[i]+=tracktor l;
                     if(type[i]==0)
                                                                  init_tracktor02_r(x_p[i],y_p[i]);
                     {
                                                                                           continue;
                          earth_cover01(x_p[i],
y_p[i]);
                                                                                      }
                          earth_cover01(x_p[i],
                                                                                 }
                                                                                 if(type[i]==1)
y_p[i] -7);
                     }
                                                                                 {
                     if(type[i]==1)
                                                                                      earth_cover02(x_p[i],y_p[i]);
                                                                                      x p[i]++;
 earth_cover02(x_p[i],y_p[i]);
                                                                  init_tracktor02_r(x_p[i],y_p[i]);
 earth_cover02(x_p[i]+7,y_p[i]);
                                                                  if(x_p[i]>=des_x+(num-i-1)*distance-tracktor_l)
                     }
                                                                                      {
                if(mode==0)
                                                                  earth_cover02(x_p[i],y_p[i]);
                     return;
                                                                  earth_cover02(x_p[i]+7,y_p[i]);
                                                                                           type[i]=2;
                break;
           }
                                                                  //init_tracktor02_f(x_p[i],y_p[i]-35);
```

```
if(type[i]==1)
                          continue;
                     }
                }
           }
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
           if(count>=num)
                                                                  earth_cover02(x_p[i]-5,y_p[i]);
                break;
                                                                                     }
           }
                                                                                }
           delay(delaytime);
                                                                                if(mode==0)
      }
                                                                                {
    }
                                                                                     return;
    //start_x>des_x&&start_y<des_y
                                                                                break;
             tracktor set off003(int
                                        start x,int
                                                                           for(i=0;i<num;i++)
start_y,int des_x,int des_y,int distance,int num)
                                                                                if(time[i]>0)
      int
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                                {
type[tracktor_num_max],time[tracktor_num_max
                                                                                     time[i]--;
],i;
                                                                                     continue;
      for(i=0;i<num;i++)
                                                                                }
      {
                                                                                if(type[i]==2)
           type[i]=0;
                                                                                {
           time[i]=i*distance;
                                                                                      count++;
                                                                                      continue;
 x p[i]=(2*start x+tra start I)/2-tracktor w/2;
                                                                                }
           y_p[i]=start_y+tra_start_d+1;
                                                                                if(type[i]==0)
      }
      while(1)
                                                                                      earth_cover01(x_p[i],y_p[i]);
      {
                                                                                     y_p[i]++;
           int count = 0,re=0;
                                                                  init_tracktor02_b(x_p[i],y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                                     if(y_p[i]>=des_y)
                                                                                     {
 re=pressed anime(x start,y start,x start,y start
);
                                                                  earth_cover01(x_p[i],y_p[i]);
           if(re!=0)
           {
                                                                  earth_cover01(x_p[i],y_p[i]+5);
                for(i=0;i<num;i++)
                                                                                          type[i]=1;
                {
                                                                                          x_p[i]+=tracktor_l;
                     if(type[i]==0)
                     {
                                                                  init_tracktor02_l(x_p[i],y_p[i]);
                                                                                          continue;
                          earth_cover01(x_p[i],
y_p[i]);
                                                                                     }
                                                                                }
                          earth_cover01(x_p[i],
y_p[i] + 5);
                                                                                if(type[i]==1)
                     }
                                                                                {
```

```
earth_cover02(x_p[i]+5,y_p[i]);
                     x_p[i]--;
                                                                  re=pressed_anime(x_start,y_start,x_start,y_start
                                                                 );
 init_tracktor02_l(x_p[i],y_p[i]);
                                                                            if(re!=0)
                                                                            {
 if(x_p[i]<=des_x+i*distance+tracktor_l)
                                                                                 for(i=0;i<num;i++)
                     {
                                                                                 {
                                                                                      if(type[i]==0)
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                                      {
                                                                                           earth_cover01(x_p[i],
 earth_cover02(x_p[i]-5,y_p[i]);
                                                                 y_p[i]);
                                                                                           earth_cover01(x_p[i],
                          type[i]=2;
                                                                 y_p[i] -7);
 //init_tracktor02_f(x_p[i]-50,y_p[i]-35);
                          continue;
                                                                                      if(type[i]==1)
                     }
                                                                                      {
                }
           }
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
           if(count>=num)
           {
                                                                  earth_cover02(x_p[i]-5,y_p[i]);
                break;
           }
                                                                                 }
           delay(delaytime);
                                                                                 if(mode==0)
      }
                                                                                 {
     }
                                                                                      return;
                                                                                 }
     //start_x>=des_x&&start_y>=des_y
                                                                                 break;
             tracktor_set_off004(int
                                        start_x,int
start_y,int des_x,int des_y,int distance,int num)
                                                                            for(i=0;i<num;i++)
     {
                                                                            {
      int
                                                                                 if(time[i]>0)
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                                 {
type[tracktor_num_max],time[tracktor_num_max
                                                                                      time[i]--;
],i;
                                                                                      continue;
      for(i=0;i<num;i++)
                                                                                 }
      {
                                                                                 if(type[i]==2)
           type[i]=0;
                                                                                 {
           time[i]=i*distance;
                                                                                      count++;
                                                                                      continue;
 x_p[i]=(2*start_x+tra_start_l)/2-tracktor_w/2;
           y_p[i]=start_y-tra_start_d-1-tracktor_l;
                                                                                 if(type[i]==0)
      }
                                                                                 {
      while(1)
                                                                                      earth_cover01(x_p[i],y_p[i]);
                                                                                      y_p[i]--;
           int count = 0,re=0;
                                                                  init_tracktor02_f(x_p[i],y_p[i]);
```

newmouse(&MouseX,&MouseY,&press);

```
if(y_p[i]<=des_y)
                                                                  tracktor_return001(start_x,start_y,des_x,des_y,d
                                                                 istance, num);
 earth_cover01(x_p[i],y_p[i]);
                                                                        else if(start x>des x&&start y>des y)
 earth_cover01(x_p[i],y_p[i]-7);
                          type[i]=1;
                                                                  tracktor_return002(start_x,start_y,des_x,des_y,d
                          x_p[i]+=tracktor_l;
                                                                 istance, num);
 init_tracktor02_r(x_p[i],y_p[i]);
                                                                       }
                                                                       else if(start_x<des_x&&start_y<des_y)
                          continue;
                     }
                }
                if(type[i]==1)
                                                                  tracktor_return003(start_x,start_y,des_x,des_y,d
                                                                 istance, num);
                                                                       }
 earth_cover02(x_p[i]+7,y_p[i]);
                                                                       else
                     x_p[i]--;
                                                                       {
 init_tracktor02_l(x_p[i],y_p[i]);
                                                                  tracktor_return004(start_x,start_y,des_x,des_y,d
                                                                 istance, num);
 if(x_p[i] \le des_x + i*distance + tracktor_l)
                                                                       }
                     {
                                                                      }
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                               tracktor_return001(int
                                                                                                          start x,int
                                                                 start y,int des x,int des y,int distance,int num)
 earth_cover02(x_p[i]-5,y_p[i]);
                                                                      {
                          type[i]=2;
                                                                       int
                                                                 x_p[tracktor_num_max],y_p[tracktor_num_max],
 //init_tracktor02_f(x_p[i]-50,y_p[i]-35);
                                                                 type[tracktor_num_max],i;
                          continue;
                                                                       if(distance<tracktor I)
                     }
                                                                       {
                }
                                                                             distance=tracktor I;
           }
           if(count>=num)
                                                                       for(i=0;i<num;i++)
           {
                break;
                                                                            type[i]=0;
                                                                            x_p[i]=des_x+i*distance;
           delay(delaytime);
                                                                            y_p[i]=des_y;
      }
                                                                            earth_fill03(x_p[i],y_p[i]);
    }
                                                                            x_p[i]-=tracktor_l;
                                                                            y p[i]-=tracktor w;
              tracktor_return0(int
     void
                                         start x,int
                                                                       }
start_y,int des_x,int des_y,int distance,int num)
                                                                       while(1)
      if(start_x<des_x&&start_y>des_y)
                                                                             int count = 0,re=0;
      {
```

```
newmouse(&MouseX,&MouseY,&press);
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                  earth_cover02(x_p[i]-6,y_p[i]);
);
           if(re!=0)
           {
                                                                  x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
                for(i=0;i<num;i++)
                                                                                           type[i]=1;
                                                                                           continue;
                     if(type[i]==0)
                                                                                      }
                     {
                                                                                 }
                                                                                 if(type[i]==1)
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                                      earth_cover01(x_p[i],y_p[i]);
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                                      y_p[i]++;
                     if(type[i]==1)
                                                                  init_tracktor02_b(x_p[i],y_p[i]);
                                                                                      if(y_p[i] > = start_y-48)
                     {
 earth_cover01(x_p[i],y_p[i]);
                                                                  earth_cover01(x_p[i],y_p[i]);
 earth_cover01(x_p[i],y_p[i]+5);
                                                                  earth_cover01(x_p[i],y_p[i]+5);
                }
                                                                                           type[i]=2;
                if(mode==0)
                                                                                           continue;
                                                                                      }
                                                                                 }
                     return;
                                                                            }
                break;
                                                                            if(count>=num)
           }
           for(i=0;i<num;i++)
                                                                                 break;
           {
                                                                            delay(delaytime);
                if(type[i]==2)
                                                                       }
                     count++;
                                                                      }
                     continue;
                                                                              tracktor_return002(int
                }
                                                                                                         start x,int
                                                                 start_y,int des_x,int des_y,int distance,int num)
                if(type[i]==0)
                                                                      {
                                                                       int
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                 x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                 type[tracktor_num_max],i;
                     x_p[i]--;
                                                                       if(distance<tracktor I)
 init_tracktor02_l(x_p[i],y_p[i]);
                                                                       {
                                                                            distance=tracktor_l;
 if(x_p[i]<=((2*start_x+tra_start_I)/2-tracktor_w/
2)+50)
                                                                       for(i=0;i<num;i++)
                     {
                                                                       {
```

```
type[num-1-i]=0;
                                                                                 if(type[i]==0)
           x_p[num-1-i]=des_x+i*distance;
                                                                                 {
           y_p[num-1-i]=des_y;
           earth_fill03(x_p[i],y_p[i]);
                                                                  earth_cover02(x_p[i]+2,y_p[i]);
           x p[num-1-i]+=tracktor l;
                                                                                      x_p[i]++;
           y_p[num-1-i]-=tracktor_w;
      }
                                                                  init_tracktor02_r(x_p[i],y_p[i]);
      while(1)
      {
                                                                  if(x_p[i]>=((2*start_x+tra_start_I)/2-tracktor_w/
           int count = 0,re=0;
                                                                 2))
                                                                                      {
 newmouse(&MouseX,&MouseY,&press);
                                                                  earth_cover02(x_p[i]+6,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
);
                                                                  earth_cover02(x_p[i]-2,y_p[i]);
           if(re!=0)
                                                                  x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
           {
                for(i=0;i<num;i++)
                                                                                           type[i]=1;
                                                                                           continue;
                     if(type[i]==0)
                                                                                      }
                     {
                                                                                 }
                                                                                 if(type[i]==1)
 earth_cover02(x_p[i]+6,y_p[i]);
                                                                                 {
                                                                                      earth_cover01(x_p[i],y_p[i]);
 earth_cover02(x_p[i]-2,y_p[i]);
                                                                                      y_p[i]++;
                     }
                     if(type[i]==1)
                                                                  init_tracktor02_b(x_p[i],y_p[i]);
                                                                                      if(y_p[i] > = start_y-48)
 earth_cover01(x_p[i],y_p[i]);
                                                                  earth_cover01(x_p[i],y_p[i]);
 earth_cover01(x_p[i],y_p[i]+5);
                     }
                                                                  earth_cover01(x_p[i],y_p[i]+5);
                                                                                           type[i]=2;
                }
                if(mode==0)
                                                                                           continue;
                                                                                      }
                                                                                 }
                     return;
                                                                            }
                break;
                                                                            if(count>=num)
           }
                                                                            {
           for(i=0;i<num;i++)
                                                                                 break;
                                                                            delay(delaytime);
                if(type[i]==2)
                {
                                                                       }
                     count++;
                                                                      }
                     continue;
                }
                                                                      void
                                                                              tracktor_return003(int
                                                                                                         start x,int
```

```
start_y,int des_x,int des_y,int distance,int num)
                                                                                       return;
                                                                                  }
      int
                                                                                  break;
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                             }
                                                                             for(i=0;i<num;i++)
type[tracktor num max],i;
      if(distance<tracktor_l)
      {
                                                                                  if(type[i]==2)
           distance=tracktor_l;
                                                                                  {
      }
                                                                                       count++;
      for(i=0;i<num;i++)
                                                                                       continue;
      {
                                                                                  }
                                                                                  if(type[i]==0)
           type[i]=0;
           x_p[i]=des_x+i*distance;
                                                                                  {
           y_p[i]=des_y;
           earth_fill03(x_p[i],y_p[i]);
                                                                   earth_cover02(x_p[i]+2,y_p[i]);
           x_p[i]-=tracktor_l;
                                                                                       x_p[i]--;
           y_p[i]-=tracktor_w;
      }
                                                                   init_tracktor02_l(x_p[i],y_p[i]);
      while(1)
                                                                   if(x_p[i]<=((2*start_x+tra_start_I)/2-tracktor_w/
      {
                                                                 2)+50)
           int count = 0,re=0;
                                                                                       {
 newmouse(&MouseX,&MouseY,&press);
                                                                   earth_cover02(x_p[i]+2,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                   earth_cover02(x_p[i]-6,y_p[i]);
);
           if(re!=0)
           {
                                                                   x_p[i]=(2*start_x+tra_start_l)/2-tracktor_w/2;
                for(i=0;i<num;i++)
                                                                                            y_p[i]-=tracktor_l;
                {
                                                                                            type[i]=1;
                     if(type[i]==0)
                                                                                            continue;
                                                                                       }
                                                                                  }
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                                  if(type[i]==1)
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                                       earth_cover01(x_p[i],y_p[i]);
                                                                                       y_p[i]--;
                     if(type[i]==1)
                                                                   init_tracktor02_f(x_p[i],y_p[i]);
 earth_cover01(x_p[i],y_p[i]-7);
                                                                   if(y_p[i]<=start_y+tra_start_l-6)</pre>
 earth_cover01(x_p[i],y_p[i]+3);
                     }
                                                                   earth_cover01(x_p[i],y_p[i]-7);
                if(mode==0)
                                                                   earth_cover01(x_p[i],y_p[i]+3);
                {
                                                                                            type[i]=2;
```

```
earth_cover02(x_p[i]-2,y_p[i]);
                          continue;
                     }
                }
                                                                                     if(type[i]==1)
           }
                                                                                     {
           if(count>=num)
                                                                 earth_cover01(x_p[i],y_p[i]-7);
                break;
           }
                                                                 earth_cover01(x_p[i],y_p[i]+3);
           delay(delaytime);
      }
                                                                                }
    }
                                                                                if(mode==0)
    void
             tracktor_return004(int
                                        start_x,int
                                                                                     return;
start_y,int des_x,int des_y,int distance,int num)
                                                                                }
                                                                                break;
      int
                                                                           }
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                           for(i=0;i<num;i++)
type[tracktor_num_max],i;
      if(distance<tracktor_l)
                                                                                if(type[i]==2)
      {
                                                                                {
           distance=tracktor I;
                                                                                     count++;
      }
                                                                                     continue;
      for(i=0;i<num;i++)
      {
                                                                                if(type[i]==0)
           type[num-1-i]=0;
                                                                                {
 x_p[num-1-i]=des_x+i*distance+tracktor_l;
                                                                 earth_cover02(x_p[i]+2,y_p[i]);
           y_p[num-1-i]=des_y-tracktor_w;
                                                                                     x_p[i]++;
           earth_fill03(x_p[i],y_p[i]);
      }
                                                                 init_tracktor02_r(x_p[i],y_p[i]);
      while(1)
                                                                 if(x_p[i]>=((2*start_x+tra_start_I)/2-tracktor_w/
           int count = 0,re=0;
                                                                2))
                                                                                     {
 newmouse(&MouseX,&MouseY,&press);
                                                                 earth_cover02(x_p[i]+6,y_p[i]);
 re=pressed_anime(x_start,y_start,x_start,y_start
);
                                                                 earth_cover02(x_p[i]-2,y_p[i]);
           if(re!=0)
           {
                                                                 x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
                for(i=0;i<num;i++)
                                                                                          y_p[i]-=tracktor_w;
                                                                                          type[i]=1;
                     if(type[i]==0)
                                                                                          continue;
                     {
                                                                                     }
                                                                                }
 earth_cover02(x_p[i]+6,y_p[i]);
                                                                                if(type[i]==1)
                                                                                {
```

```
earth_cover01(x_p[i],y_p[i]);
                                                                            draw_copak(des_x[i],des_y[i]);
                                                                       }
                     y_p[i]--;
                                                                       while(1)
 init_tracktor02_f(x_p[i],y_p[i]);
                                                                       {
                                                                            int total=0,re=0;
 if(y_p[i]<=start_y+tra_start_l-6)</pre>
                     {
                                                                  newmouse(&MouseX,&MouseY,&press);
 earth_cover01(x_p[i],y_p[i]-7);
                                                                  re=pressed_anime(x_start,y_start,x_start,y_start
                                                                 );
                                                                            if(re!=0)
 earth_cover01(x_p[i],y_p[i]+3);
                          type[i]=2;
                                                                            {
                          continue;
                                                                                 for(i=0;i<num;i++)
                     }
                                                                                 {
                }
                                                                                      if(type[i]==0)
           }
                                                                                      {
           if (count >= num)
                                                                                           earth_cover01(x_p[i],
           {
                                                                 y_p[i]);
                break;
                                                                                           earth cover01(x p[i],
           }
                                                                 y_p[i] + 5);
           delay(delaytime);
                                                                                      }
      }
                                                                                      if(type[i]==1)
    }
                                                                  earth_cover02(x_p[i],y_p[i]);
    void picker anime(int start x,int start y,int
*des_x,int *des_y,int distance,int num)
                                                                  earth_cover02(x_p[i]+7,y_p[i]);
     #define cost_time 50
                                                                                 }
      int
                         x_p[tracktor_num_max],
                                                                                 if(mode==0)
y_p[tracktor_num_max],
                                                                                 {
type[tracktor_num_max],\
                                                                                      return;
                                                                                 }
time[tracktor_num_max],count[tracktor_num_ma
                                                                                 break;
x], i,co_time[tracktor_num_max],xy[2];
                                                                            }
      xy[0]=start_x,xy[1]=start_y;
                                                                            for(i=0;i<num;i++)
      // if(des_x[0] <= 0)
      // return;
                                                                                 if(des_x[i] <= 0)
      for (i = 0; i < num; i++)
                                                                                 {
      {
                                                                                      count[i]=3;
           type[i] = 0;
                                                                                 if(count[i]==0)
           count[i]=0;
           co_time[i]=0;
                                                                                 {
           time[i] = i * distance;
           x_p[i] = (2 * start_x + tra_start_l) / 2 -
                                                                  picker_set_off(x_p,y_p,start_x,start_y,des_x[i],d
                                                                 es_y[i]+12,time,i,count,type);
tracktor_w / 2;
                                                                                      continue;
           y_p[i] = start_y + tra_start_d + 1;
```

```
else if(count[i]==1)
                                                                            draw_setoff(xy);
                                                                            delay(delaytime);
                     // if(co_time[i]==0)
                                                                       }
                     //{
                                                                       return;
                     //
                                                                     }
 init_picker_f(des_x[i],des_y[i]);
                     // co_time[i]++;
                                                                     void picker_set_off(int *x_p,int *y_p,int
                          continue;
                                                                start_x,int start_y,int des_x,int des_y,int *time,int
                     //}
                                                                num,int *count,int* type)
                     if(co_time[i]<cost_time)</pre>
                                                                     {
                                                                       if (start x < des x && start y < des y)
                                                                       {
 init_picker_f(des_x[i],des_y[i]);
                                                                           picker_set_off01( x_p,y_p,des_x, des_y,
                          co_time[i]++;
                                                                time, num, count, type);
                          continue;
                                                                       }
                     }
                                                                       else if (start_x < des_x && start_y > des_y)
                     else
if(co time[i]>=cost time)
                                                                           picker_set_off02(x_p,y_p, des_x, des_y,
                                                                time, num,count,type);
                     {
                          type[i]=0;
                          count[i]++;
                                                                       else if (start_x > des_x && start_y < des_y)
                                                                       {
 x_p[i]=des_x[i],y_p[i]=des_y[i];
                                                                           picker_set_off03(x_p,y_p, des_x, des_y,
                          earth_cover01(x_p[i],
                                                                time, num, count, type);
y p[i]-7);
                                                                       }
                          earth_cover01(x_p[i],
                                                                       else
y_p[i] + 5);
                          continue;
                                                                           picker_set_off04(x_p,y_p, des_x, des_y,
                     }
                                                                time, num, count, type);
                                                                      }
                else if(count[i]==2)
                                                                     }
                                                                     // start_x<des_x&&start_y<des_y
                                                                     void picker set off01( int *x p,int *y p,int
 picker_return(x_p,y_p,start_x,start_y,des_x[i],de
                                                                des_x, int des_y, int *time, int i,int *count,int
s_y[i]-5-tracktor_l,i,count,type);
                     continue;
                                                                 *type)
                                                                     {
                else if(count[i]>=3)
                                                                       //
                                                                                          x_p[tracktor_num_max],
                {
                                                                y_p[tracktor_num_max],
                     total++;
                                                                type[tracktor_num_max],
                     continue;
                                                                time[tracktor num max], i;
                                                                       // for (i = 0; i < num; i++)
                }
           }
                                                                       //{
           if(total==num)
                                                                       // type[i] = 0;
           {
                                                                       // time[i] = i * distance;
                                                                       // x_p[i] = (2 * start_x + tra_start_l) / 2 -
                break;
```

```
x_p[i] += tracktor_l;
tracktor_w / 2;
                                                                                init_picker_r(x_p[i], y_p[i]);
      // y_p[i] = start_y + tra_start_d + 1;
      //}
                                                                           }
      // //clrmous(MouseX,MouseY);
                                                                           return;
      // while (1)
      //{
                                                                      if (type[i] == 1)
      // int re=0;
      // newmouse(&MouseX,&MouseY,&press);
                                                                           earth_cover02(x_p[i], y_p[i]);
                                                                           x_p[i]++;
                                                                           init picker r(x p[i], y p[i]);
re=pressed anime(x start,y start,x start,y start);
                                                                           if (x_p[i] >= des_x-tracktor_l)
      // if(re!=0)
      //{
                                                                                earth_cover02(x_p[i],y_p[i]);
      //
           if(type[i]==0)
                                                                                earth_cover02(x_p[i]+7,y_p[i]);
      // {
      //
                earth_cover01(x_p[i], y_p[i]);
                                                                                //x_p[i]=des_x,y_p[i]=des_y;
      //
                earth\_cover01(x_p[i], y_p[i] + 5);
                                                                                type[i]=2;
      //
           }
           if(type[i]==1)
                                                                 //init_tracktor01_f(x_p[i],y_p[i]-35);
      //
      // {
      //
                earth_cover02(x_p[i],y_p[i]);
                                                                           return;
      //
                earth cover02(x p[i]+7,y p[i]);
                                                                      }
      //
                                                                      // if (count >= num)
          }
           if(mode==0)
                                                                      //{
                                                                      //
      //
          {
                                                                           break;
      //
                return;
                                                                      //}
                                                                      //}
      //
          }
      //}
                                                                     }
      if (time[i] > 0)
                                                                     // start_x<des_x&&start_y>des_y
           time[i]--;
                                                                     void picker_set_off02(int *x_p,int *y_p, int
                                                                des_x, int des_y, int *time, int i,int *count,int*
           return;
      }
                                                                type)
      if (type[i] == 2)
                                                                     {
                                                                      //
                                                                                         x_p[tracktor_num_max],
      {
                                                                               int
           count[i]++;
                                                                y p[tracktor num max],
           return;
                                                                type[tracktor_num_max],
      }
                                                                time[tracktor_num_max], i;
      if (type[i] == 0)
                                                                      // for (i = 0; i < num; i++)
                                                                      //{
           earth_cover01(x_p[i], y_p[i]);
                                                                      // type[i] = 0;
           y_p[i]++;
                                                                      // time[i] = i * distance;
           init_picker_b(x_p[i], y_p[i]);
                                                                      // x_p[i] = (2 * start_x + tra_start_l) / 2 -
           if (y_p[i] >= des_y)
                                                                tracktor_w / 2;
           {
                                                                      // y_p[i] = start_y - tra_start_d - 1 -
                earth_cover01(x_p[i], y_p[i]);
                                                                tracktor_l;
                earth_cover01(x_p[i], y_p[i] + 5);
                                                                      //}
                type[i] = 1;
                                                                      // //clrmous(MouseX,MouseY);
```

```
// while (1)
      //{
                                                                      if (type[i] == 1)
      // int re=0;
      // newmouse(&MouseX,&MouseY,&press);
                                                                            earth_cover02(x_p[i], y_p[i]);
                                                                           x p[i]++;
re=pressed_anime(x_start,y_start,x_start,y_start);
                                                                           init_picker_r(x_p[i], y_p[i]);
                                                                           if (x_p[i] >= des_x - tracktor I)
      // if(re!=0)
      //{
                                                                           {
      // if(type[i]==0)
                                                                                earth_cover02(x_p[i],y_p[i]);
      // {
                                                                                earth cover02(x p[i]+7,y p[i]);
      //
                earth_cover01(x_p[i], y_p[i]);
                                                                                //x_p[i]=des_x,y_p[i]=des_y;
      //
                earth_cover01(x_p[i], y_p[i] -7);
                                                                                type[i]=2;
      //
          }
           if(type[i]==1)
      //
                                                                 //init_tracktor01_f(x_p[i],y_p[i]-35);
      //
      //
                earth_cover02(x_p[i],y_p[i]);
                                                                           return;
      //
                earth_cover02(x_p[i]+7,y_p[i]);
                                                                      }
                                                                      // if (count >= num)
      //
                                                                      //{
      //
           if(mode==0)
                                                                      // break;
      //
      //
                return;
                                                                      //}
      // }
                                                                     }
      //}
      if (time[i] > 0)
                                                                     // start_x>des_x&&start_y<des_y
                                                                     void picker_set_off03(int *x_p,int *y_p, int
                                                                des x, int des y, int *time, int i,int *count,int*
           time[i]--;
           return;
                                                                type)
                                                                     {
      if (type[i] == 2)
                                                                      //
                                                                                          x_p[tracktor_num_max],
                                                                                int
      {
                                                                y_p[tracktor_num_max],
                                                                type[tracktor_num_max],
           count[i]++;
           return;
                                                                time[tracktor_num_max], i;
      }
                                                                      // for (i = 0; i < num; i++)
      if (type[i] == 0)
                                                                      //{
                                                                      // type[i] = 0;
           earth_cover01(x_p[i], y_p[i]);
                                                                      // time[i] = i * distance;
           y_p[i]--;
                                                                      // x_p[i] = (2 * start_x + tra_start_l) / 2 -
           init_picker_f(x_p[i], y_p[i]);
                                                                tracktor_w / 2;
           if (y_p[i] \le des_y)
                                                                      // y_p[i] = start_y + tra_start_d + 1;
           {
                                                                      //}
                earth_cover01(x_p[i], y_p[i]);
                                                                      // //clrmous(MouseX,MouseY);
                earth_cover01(x_p[i], y_p[i] - 7);
                                                                      // while (1)
                                                                      //{
                type[i] = 1;
                x_p[i] += tracktor_l;
                                                                      // int re=0;
                init_picker_r(x_p[i], y_p[i]);
                                                                      // newmouse(&MouseX,&MouseY,&press);
           }
                                                                      //
           return;
                                                                re=pressed_anime(x_start,y_start,x_start,y_start);
```

```
// if(re!=0)
                                                                      if (x_p[i] <= des_x +tracktor_l)</pre>
//{
     if(type[i]==0)
                                                                           earth_cover02(x_p[i]+2,y_p[i]);
//
                                                                           earth_cover02(x_p[i]-5,y_p[i]);
   {
//
                                                                           x_p[i]=des_x,y_p[i]=des_y;
          earth_cover01(x_p[i], y_p[i]);
//
          earth_cover01(x_p[i], y_p[i] + 5);
                                                                           type[i]=2;
//
     }
//
     if(type[i]==1)
                                                            //init_tracktor01_f(x_p[i]-50,y_p[i]-35);
//
    {
                                                                      }
//
          earth cover02(x p[i]+2,y p[i]);
                                                                      return;
//
          earth_cover02(x_p[i]-5,y_p[i]);
                                                                 }
                                                                 // if (count >= num)
//
     }
     if(mode==0)
                                                                 //{
                                                                      break;
//
    {
                                                                 //
//
          return;
                                                                //}
//
                                                               }
     }
//}
                                                               // start_x>=des_x&&start_y>=des_y
if (time[i] > 0)
                                                               void picker set off04( int *x p,int *y p,int
                                                          des_x, int des_y, int *time, int i,int *count,int*
     time[i]--;
     return;
                                                          type)
}
                                                               {
if (type[i] == 2)
                                                                 //
                                                                          int
                                                                                    x_p[tracktor_num_max],
{
                                                          y_p[tracktor_num_max],
     count[i]++;
                                                          type[tracktor_num_max],
     return;
                                                          time[tracktor num max], i;
}
                                                                 // for (i = 0; i < num; i++)
if (type[i] == 0)
                                                                 //{
                                                                 // type[i] = 0;
     earth_cover01(x_p[i], y_p[i]);
                                                                 // time[i] = i * distance;
                                                                 // x_p[i] = (2 * start_x + tra_start_l) / 2 -
     y p[i]++;
     init_picker_b(x_p[i], y_p[i]);
                                                          tracktor_w / 2;
     if (y_p[i] >= des_y)
                                                                 // y_p[i] = start_y - tra_start_d - 1 -
     {
                                                          tracktor_l;
          earth_cover01(x_p[i], y_p[i]);
                                                                 //}
                                                                 // //clrmous(MouseX,MouseY);
          earth\_cover01(x_p[i], y_p[i] + 5);
          type[i] = 1;
                                                                 // while (1)
          x p[i] += tracktor l;
                                                                 //{
          init_picker_l(x_p[i], y_p[i]);
                                                                 // int count = 0,re=0;
     }
                                                                 // newmouse(&MouseX,&MouseY,&press);
     return;
}
                                                          re=pressed anime(x start,y start,x start,y start);
if (type[i] == 1)
                                                                 // if(re!=0)
                                                                 //{
     earth_cover02(x_p[i] + 5, y_p[i]);
                                                                 //
                                                                      if(type[i]==0)
     x p[i]--;
                                                                 //
     init_picker_l(x_p[i], y_p[i]);
                                                                 //
                                                                           earth_cover01(x_p[i], y_p[i]);
```

```
//
           earth_cover01(x_p[i], y_p[i] -7);
                                                                       type[i]=2;
 //
      if(type[i]==1)
                                                             //init_tracktor01_f(x_p[i]-50,y_p[i]-35);
 //
     {
                                                                  }
 //
           earth_cover02(x_p[i]+2,y_p[i]);
                                                                  return;
                                                                 }
 //
           earth_cover02(x_p[i]-5,y_p[i]);
 //
                                                                 //
                                                                       if (count >= num)
      }
 //
      if(mode==0)
                                                                 //
                                                                       {
 //
      {
                                                                 //
                                                                            break;
 //
                                                                 //
           return;
 //
      }
                                                                 //
                                                                       delay(delaytime);
 //}
                                                                 //}
 if (time[i] > 0)
                                                                 }
      time[i]--;
                                                                 void picker_return(int *x_p,int *y_p,int
      return;
                                                            start_x,int start_y,int des_x,int des_y,int num,int
                                                            *count,int* type)
 }
 if (type[i] == 2)
                                                                 if(start x<des x&&start y>des y)
      count[i]++;
      return;
                                                                  picker_return01(x_p,y_p,start_x,start_y
 }
                                                            num,count,type);
 if (type[i] == 0)
                                                                 }
                                                                 else if(start_x>des_x&&start_y>des_y)
 earth_cover01(x_p[i], y_p[i]);
 y p[i]--;
                                                                   picker_return02( x_p,y_p,start_x,start_y ,
 init_picker_f(x_p[i], y_p[i]);
                                                            num,count,type);
 if (y_p[i] \le des_y)
 {
                                                                 else if(start_x<des_x&&start_y<des_y)
      earth_cover01(x_p[i], y_p[i]);
      earth_cover01(x_p[i], y_p[i] - 7);
                                                                  picker_return03(x_p,y_p,start_x,start_y
      type[i] = 1;
                                                            num,count,type);
      x_p[i] += tracktor_l;
                                                                 }
      init_picker_r(x_p[i], y_p[i]);
                                                                 else
 }
 return;
                                                                   picker_return04(x_p,y_p,
                                                            start_x,start_y ,num,count,type);
                                                                 }
if (type[i] == 1)
                                                                 }
 earth_cover02(x_p[i] + 7, y_p[i]);
 x p[i]--;
                                                                 void picker_return01( int *x_p,int *y_p,int
 init_picker_l(x_p[i], y_p[i]);
                                                            start x,int start y, int i,int *count,int *type)
 if (x_p[i] <= des_x + tracktor_l)</pre>
                                                                 {
                                                                                                             int
 {
      earth_cover02(x_p[i]+2,y_p[i]);
                                                            x_p[tracktor_num_max],y_p[tracktor_num_max],
      earth_cover02(x_p[i]-5,y_p[i]);
                                                            type[tracktor_num_max],i;
      x_p[i]=des_x,y_p[i]=des_y;
                                                                 // if(distance<tracktor I)
```

```
//{
                                                                       if(type[i]==2)
    //
           distance=tracktor_l;
                                                                       {
    //}
                                                                            count[i]++;
    // for(i=0;i<num;i++)
                                                                            return;
    //{
    //
                                                                       if(type[i]==0)
           type[i]=0;
           x_p[i]=des_x+i*distance;
    //
                                                                       {
    //
           y_p[i]=des_y;
                                                                            earth_cover02(x_p[i]+2,y_p[i]);
    //
           earth_fill03(x_p[i],y_p[i]);
                                                                            x_p[i]--;
    //
                                                                            init_picker01_l(x_p[i],y_p[i]);
           x p[i]-=tracktor l;
    //
           y_p[i]-=tracktor_w;
                                                                  if(x_p[i]<=((2*start_x+tra_start_l)/2-tracktor w/
    //}
    // while(1)
                                                                2)+50)
    //{
                                                                            {
    //
           int count = 0,re=0;
                                                                                 earth_cover02(x_p[i]+2,y_p[i]);
    //
                                                                                 earth_cover02(x_p[i]-6,y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                  x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                                 type[i]=1;
                                                                            }
);
    //
           if(re!=0)
                                                                            return;
    //
                                                                       }
           {
    //
                for(i=0;i<num;i++)
                                                                       if(type[i]==1)
    //
                {
    //
                     if(type[i]==0)
                                                                            earth_cover01(x_p[i],y_p[i]);
    //
                     {
                                                                            y p[i]++;
    //
                                                                            init_picker01_b(x_p[i],y_p[i]);
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                            if(y_p[i] > = start_y-48)
    //
                                                                            {
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                                 earth_cover01(x_p[i],y_p[i]);
                                                                                 earth_cover01(x_p[i],y_p[i]+5);
    //
    //
                     if(type[i]==1)
                                                                                 type[i]=2;
    //
                     {
                                                                            }
    //
                                                                            return;
 earth_cover01(x_p[i],y_p[i]);
                                                                       }
                                                                            // if(count>=num)
    //
                                                                            //{
 earth_cover01(x_p[i],y_p[i]+5);
    //
                     }
                                                                            // break;
    //
                }
                                                                            //}
    //
                if(mode==0)
                                                                            // delay(delaytime);
    //
                                                                     }
    //
                     return;
    //
                                                                     void picker_return02(int *x_p,int *y_p,int
    //
                break;
                                                                start_x,int start_y, int i,int *count,int *type)
    //
           }
                                                                     {
      // for(i=0;i<num;i++)
                                                                       //
                                                                                                                 int
      //{
                                                                x_p[tracktor_num_max],y_p[tracktor_num_max],
```

```
for(i=0;i<num;i++)
type[tracktor_num_max],i;
                                                                      //
      // if(distance<tracktor_I)
                                                                     //
      //{
                                                                      if(type[i]==2)
      //
           distance=tracktor_l;
                                                                     {
      //}
                                                                           count[i]++;
      // for(i=0;i<num;i++)
                                                                          return;
      //{
                                                                      }
      // type[num-1-i]=0;
                                                                      if(type[i]==0)
      // x_p[num-1-i]=des_x+i*distance;
      // y p[num-1-i]=des y;
                                                                           earth_cover02(x_p[i]+2,y_p[i]);
      // earth_fill03(x_p[i],y_p[i]);
                                                                          x_p[i]++;
                                                                          init_picker01_r(x_p[i],y_p[i]);
      // x_p[num-1-i]+=tracktor_l;
      // y_p[num-1-i]-=tracktor_w;
      //}
                                                                 if(x_p[i]>=((2*start_x+tra_start_I)/2-tracktor_w/
      // while(1)
                                                               2))
      //{
                                                                          {
      //
           int count = 0,re=0;
                                                                               earth_cover02(x_p[i]+6,y_p[i]);
      //
                                                                               earth_cover02(x_p[i]-2,y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                 x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
      //
 re=pressed anime(x start,y start,x start,y start
                                                                               type[i]=1;
                                                                          }
);
      //
           if(re!=0)
                                                                          return;
      //
                                                                      }
           {
      //
                for(i=0;i<num;i++)
                                                                      if(type[i]==1)
      //
                {
      //
                     if(type[i]==0)
                                                                           earth_cover01(x_p[i],y_p[i]);
      //
                     {
                                                                          y_p[i]++;
                                                                          init_picker01_b(x_p[i],y_p[i]);
      //
 earth_cover02(x_p[i]+6,y_p[i]);
                                                                          if(y_p[i] > = start_y-48)
      //
                                                                          {
 earth_cover02(x_p[i]-2,y_p[i]);
                                                                               earth_cover01(x_p[i],y_p[i]);
      //
                                                                               earth_cover01(x_p[i],y_p[i]+5);
                     }
      //
                     if(type[i]==1)
                                                                               type[i]=2;
      //
                                                                          }
      //
                                                                          return;
 earth_cover01(x_p[i],y_p[i]);
                                                                     }
                                                                          //}
 earth_cover01(x_p[i],y_p[i]+5);
                                                                     //
                                                                          if(count>=num)
      //
                     }
                                                                     //
                                                                          {
      //
                                                                      //
                                                                               break;
      //
                if(mode==0)
                                                                      //
      //
                                                                     //
                                                                          delay(delaytime);
      //
                     return;
                                                                     //}
      //
                                                                    }
      //
                break;
      //
           }
                                                                    void picker_return03(int *x_p,int *y_p,int
```

```
//
start_x,int start_y, int i,int *count,int *type)
                                                                                     return;
                                                                      //
                                                                                }
      //
                                                                      //
                                                                                break;
                                                int
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                      //
type[tracktor num max],i;
                                                                           for(i=0;i<num;i++)
      // if(distance<tracktor_l)
                                                                      //
                                                                           {
                                                                      if(type[i]==2)
      //{
      //
           distance=tracktor_l;
                                                                      {
      //}
                                                                           count[i]++;
      // for(i=0;i<num;i++)
                                                                           return;
      //{
                                                                      }
                                                                      if(type[i]==0)
      // type[i]=0;
      // x_p[i]=des_x+i*distance;
                                                                      {
      // y_p[i]=des_y;
                                                                            earth_cover02(x_p[i]+2,y_p[i]);
      // earth_fill03(x_p[i],y_p[i]);
                                                                           x_p[i]--;
 // x_p[i]-=tracktor_l;
                                                                           init_picker01_l(x_p[i],y_p[i]);
      // y_p[i]-=tracktor_w;
      //}
                                                                 if(x_p[i]<=((2*start_x+tra_start_l)/2-tracktor_w/
      // while(1)
                                                                2)+50)
      //{
                                                                           {
      //
           int count = 0,re=0;
                                                                                earth cover02(x p[i]+2,y p[i]);
      //
                                                                                earth_cover02(x_p[i]-6,y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                 x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
      //
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                                y_p[i]-=tracktor_l;
                                                                                type[i]=1;
);
      //
           if(re!=0)
                                                                           }
      //
           {
                                                                           return;
      //
                for(i=0;i<num;i++)
                                                                      }
      //
                {
                                                                      if(type[i]==1)
      //
                     if(type[i]==0)
                                                                      {
      //
                     {
                                                                            earth_cover01(x_p[i],y_p[i]);
      //
                                                                           y_p[i]--;
 earth_cover02(x_p[i]+2,y_p[i]);
                                                                           init_picker01_f(x_p[i],y_p[i]);
                                                                           if(y_p[i]<=start_y+tra_start_I-6)</pre>
 earth_cover02(x_p[i]-6,y_p[i]);
                                                                           {
      //
                     }
                                                                                earth_cover01(x_p[i],y_p[i]-7);
      //
                     if(type[i]==1)
                                                                                earth_cover01(x_p[i],y_p[i]+3);
      //
                     {
                                                                                type[i]=2;
      //
                                                                           }
 earth_cover01(x_p[i],y_p[i]-7);
                                                                           return;
                                                                      }
      //
 earth_cover01(x_p[i],y_p[i]+3);
                                                                      //
                                                                           }
      //
                     }
                                                                      //
                                                                           if(count>=num)
      //
                                                                      //
      //
                if(mode==0)
                                                                      //
                                                                                break;
      //
                {
                                                                      //
                                                                           }
```

```
//
      //
           delay(delaytime);
                                                                                     }
      //}
                                                                      //
                                                                                 }
    }
                                                                       //
                                                                                 if(mode==0)
                                                                      //
                                                                                 {
                                                                       //
     void picker_return04(int *x_p,int *y_p,int
                                                                                      return;
start_x,int start_y, int i,int *count,int *type)
                                                                       //
                                                                                 break;
    {
                                                                      //
      //
                                                int
                                                                       //
x_p[tracktor_num_max],y_p[tracktor_num_max],
                                                                           for(i=0;i<num;i++)
type[tracktor num max],i;
                                                                       if(type[i]==2)
      // if(distance<tracktor_l)</pre>
      //{
      //
           distance=tracktor_l;
                                                                            count[i]++;
      //}
                                                                            return;
      // for(i=0;i<num;i++)
      //{
                                                                       if(type[i]==0)
      //
           type[num-1-i]=0;
      //
                                                                            earth_cover02(x_p[i]+2,y_p[i]);
 x_p[num-1-i]=des_x+i*distance+tracktor I;
                                                                            x p[i]++;
                                                                            init\_picker01\_r(x\_p[i],y\_p[i]);
      // y_p[num-1-i]=des_y-tracktor_w;
      //
           earth_fill03(x_p[i],y_p[i]);
      //}
                                                                  if(x_p[i]>=((2*start_x+tra_start_I)/2-tracktor_w/
      // while(1)
                                                                2))
      //{
                                                                           {
      //
           int count = 0,re=0;
                                                                                 earth_cover02(x_p[i]+6,y_p[i]);
      //
                                                                                 earth_cover02(x_p[i]-2,y_p[i]);
 newmouse(&MouseX,&MouseY,&press);
                                                                  x_p[i]=(2*start_x+tra_start_I)/2-tracktor_w/2;
 re=pressed_anime(x_start,y_start,x_start,y_start
                                                                                y_p[i]-=tracktor_w;
);
                                                                                 type[i]=1;
      //
           if(re!=0)
                                                                           }
      //
          {
                                                                            return;
      //
                for(i=0;i<num;i++)
      //
                                                                       if(type[i]==1)
                {
      //
                     if(type[i]==0)
                                                                            earth_cover01(x_p[i],y_p[i]);
      //
                     {
      //
                                                                           y_p[i]--;
 earth_cover02(x_p[i]+6,y_p[i]);
                                                                            init_picker01_f(x_p[i],y_p[i]);
      //
                                                                            if(y_p[i]<=start_y+tra_start_l-6)</pre>
 earth_cover02(x_p[i]-2,y_p[i]);
                                                                           {
      //
                                                                                 earth_cover01(x_p[i],y_p[i]-7);
      //
                                                                                 earth_cover01(x_p[i],y_p[i]+3);
                     if(type[i]==1)
      //
                                                                                 type[i]=2;
                     {
                                                                           }
                                                                           return;
 earth_cover01(x_p[i],y_p[i]-7);
      //
                                                                       }
 earth_cover01(x_p[i],y_p[i]+3);
                                                                       //
                                                                           }
```

```
//
                                                                        line(300, 130, 420, 130);
              //
                                                                        rectangle(310, 130, 410, 200);
                        break;
              //
                                                                        puthz(330, 150, "仓库管理", 16, 16, BLUE);
                   delay(delaytime);
              //
              //}
                                                                        quit();
        }
                                                                        // last();
                                                                   }
13 WELCOME.C
    #include "COMMON.H"
                                                                   // draw the tractor in welcome page
    #include "WELCOME.H"
                                                                   void draw_wel_tractors()
    #include "mouse.h"
    #include "PARAMETE.H"
                                                                        // The wheels
    #include "HOME.H"
                                                                        int i, d;
    #include "LOGFUN.H"
                                                                        setcolor(BROWN);
    /*void main()
                                                                        setfillstyle(1, BLUE);
    {
                                                                        bar(94, 185, 100, 215);
                                                                        bar(200, 185, 206, 215);
         int gdriver, gmode;
         gdriver=DETECT;
                                                                        bar(94, 265, 100, 295);
         initgraph(&gdriver,&gmode,"..\\BORLANDC\\BGI
                                                                        bar(200, 265, 206, 295);
");
                                                                        setfillstyle(1, RED);
         draw_wel();
         closegraph();
                                                                        // The rectangle of the machine
    }*/
                                                                        bar(100, 150, 200, 310);
                                                                        rectangle(105, 155, 195, 305);
    // draw the buttons in welcome page
                                                                        setfillstyle(1, YELLOW);
    void draw_wel_buttons(void)
                                                                        setlinestyle(0, 0, 1);
    {
         clrmous(MouseX, MouseY);
                                                                        // The small bar
         cleardevice();
                                                                        for (i = 0, d = 80; i < 8; i++)
         setbkcolor(WHITE);
                                                                        {
         puthz(180, 30, "棉花采摘模拟系统", 32, 32,
                                                                             bar(d - 2, 125, d + 2, 155);
BLUE);
                                                                             rectangle(d - 3, 125, d + 3, 155);
         setlinestyle(0, 0, 1);
                                                                             d += 20;
         setcolor(DARKGRAY);
                                                                        }
                                                                        setlinestyle(0, 0, 3);
                                                                        bar(80, 130, 220, 150);
         setfillstyle(1, 13);
         rectangle(450, 110, 550, 170);
                                                                        rectangle(80, 130, 220, 150);
         puthz(470, 130, "编辑参数", 16, 16, BLUE);
                                                                        for (i = 0, d = 100; i < 6; i++)
         rectangle(450, 210, 550, 270);
                                                                        {
         puthz(470, 230, "开始模拟", 16, 16, BLUE);
                                                                             line(d, 130, d, 150);
         rectangle(450, 310, 550, 370);
                                                                             d += 20;
         puthz(470, 330, "参数列表", 16, 16, BLUE);
                                                                        }
         rectangle(450, 410, 550, 470);
                                                                        setfillstyle(1, RED);
         puthz(460, 430, "帮助及说明", 16, 16, BLUE);
                                                                        bar(130, 120, 170, 160);
                                                                        rectangle(130, 120, 170, 160);
         line(300, 130, 360, 100);
                                                                   }
```

//

if(count>=num)

line(360, 100, 420, 130);

```
void enter_next()
// draw the cotton field in welcome page
                                                                {
void draw wel cofield()
                                                                     static int flag = 0, flag1 = 1;
                                                                     int dian[8] = {300, 130, 360, 100, 420, 130, 300,
     // The cotton field
                                                           130};
                                                                     if (mouse_press(450, 110, 550, 170) == 0 ||
     int i, d;
     setcolor(BROWN);
                                                           mouse press(450,
                                                                                 210,
                                                                                         550,
                                                                                                 270)
                                                                                                                    Ш
     rectangle(50, 100, 300, 400);
                                                           mouse_press(450,
                                                                                 310,
                                                                                         550,
                                                                                                 370)
                                                                                                               0
                                                                                                                    Ш
     setlinestyle(0, 0, 3);
                                                           mouse_press(450, 410, 550, 470) == 0 ||
     for (d = 58; d <= 300; d += 20)
                                                                          mouse press(310, 130, 410, 200) == 0 ||
     {
                                                           mouse_press(0, 0, 40, 30) == 0 || mouse_press(0, 450, 40,
          for (i = 100; i \le 400; i += 2)
                                                           480) == 0
         {
                                                                     {
               int temp = rand() % 2;
                                                                          MouseS = 0;
               line(d + temp, i, d + temp, i);
                                                                     }
         }
                                                                     if (mouse_press(450, 110, 550, 170) == 2 ||
     }
                                                           mouse_press(450,
                                                                                 210,
                                                                                         550,
                                                                                                 270)
                                                                                                               2
                                                                                                                    Ш
     for (i = 0; i < 1000; i++)
                                                           mouse_press(450,
                                                                                 310,
                                                                                         550,
                                                                                                 370)
                                                                                                               2
                                                                                                                    Ш
                                                                                                         ==
                                                           mouse press(450, 410, 550, 470) == 2 | |
          int t1 = rand() \% 250, t2 = rand() \% 300;
                                                                          mouse_press(310, 130, 410, 200) == 2 ||
          line(t1 + 50, t2 + 100, t1 + 50, t2 + 100);
                                                           mouse press(0, 0, 40, 30) == 2 | | mouse press(0, 450, 40,
                                                           480) == 2)
     setfillstyle(1, BROWN);
                                                                     {
     bar(78, 130, 222, 400);
                                                                          MouseS = 1;
     setcolor(WHITE);
                                                                     }
     for (i = 0; i < 100; i++)
                                                                     if (mouse_press(450, 110, 550, 170) == 2)
     {
          int t1 = rand() % 144, t2 = rand() % 270;
          line(t1 + 78, t2 + 130, t1 + 78, t2 + 130);
                                                                          if (flag1 == 1)
     }
                                                                          {
     /*setfillstyle(1,BROWN);
                                                                               clrmous(MouseX, MouseY);
     bar(50,100,300,400);
                                                                               setfillstyle(1, CYAN);
     setlinestyle(0,0,3);
                                                                               bar(450, 110, 550, 170);
     setcolor(WHITE);
                                                                               puthz(470, 130, "编辑参数", 16, 16,
     for(i=50;i<=300;i+=6)
                                                           BLUE);
     {
          for(d=100;d<=400;d+=2)
                                                                               flag = 0;
                                                                               flag1 = 0;
         {
               int temp=rand()%2;
                                                                          }
               line(i+temp,d,i+temp,d);
                                                                     }
               //temp=rand()%2;
                                                                     else if (mouse_press(450, 210, 550, 270) == 2)
               //line(i+temp,d,i+temp,d);
         }
                                                                          if (flag1 == 1)
     }*/
                                                                          {
                                                                               clrmous(MouseX, MouseY);
                                                                               setfillstyle(1, CYAN);
                                                                               bar(450, 210, 550, 270);
// enter the edit page
```

{

}

```
puthz(470, 230, "开始模拟", 16, 16,
                                                                         else if (flag == 0)
BLUE);
                   flag = 0;
                                                                              clrmous(MouseX, MouseY);
                   flag1 = 0;
                                                                              setfillstyle(1, 0);
              }
                                                                              bar(450, 110, 550, 170);
         }
                                                                              bar(450, 210, 550, 270);
         else if (mouse press(450, 310, 550, 370) == 2)
                                                                              bar(450, 310, 550, 370);
                                                                              bar(450, 410, 550, 470);
              if (flag1 == 1)
              {
                                                                              rectangle(450, 110, 550, 170);
                                                                              puthz(470, 130, "编辑参数", 16, 16, BLUE);
                   clrmous(MouseX, MouseY);
                   setfillstyle(1, CYAN);
                                                                              rectangle(450, 210, 550, 270);
                   bar(450, 310, 550, 370);
                                                                              puthz(470, 230, "开始模拟", 16, 16, BLUE);
                   puthz(470, 330, "参数列表", 16, 16,
                                                                              rectangle(450, 310, 550, 370);
BLUE);
                                                                              puthz(470, 330, "参数列表", 16, 16, BLUE);
                   flag = 0;
                                                                              rectangle(450, 410, 550, 470);
                   flag1 = 0;
                                                                              puthz(460, 430, "帮助及说明", 16, 16,
              }
                                                               BLUE);
         }
                                                                              setfillstyle(1, 0);
         else if (mouse_press(450, 410, 550, 470) == 2)
                                                                              bar(300, 100, 420, 200);
         {
                                                                              puthz(330, 150, "仓库管理", 16, 16, BLUE);
                                                                              rectangle(310, 130, 410, 200);
              if (flag1 == 1)
              {
                                                                              line(300, 130, 360, 100);
                   clrmous(MouseX, MouseY);
                                                                              line(360, 100, 420, 130);
                   setfillstyle(1, CYAN);
                                                                              line(300, 130, 420, 130);
                   bar(450, 410, 550, 470);
                   puthz(460, 430, "帮助及说明", 16, 16,
                                                                             flag = 1;
BLUE);
                                                                             flag1 = 1;
                   flag = 0;
                                                                         }
                   flag1 = 0;
              }
                                                                         // Enter the edit page
         }
                                                                         if (mouse_press(450, 110, 550, 170) == 1)
         else if (mouse_press(310, 130, 410, 200) == 2)
                                                                         {
         {
                                                                             mode = 2;
              if (flag1 == 1)
                                                                         }
              {
                   clrmous(MouseX, MouseY);
                                                                         // Enter the simulation page
                                                                         if (mouse press(450, 210, 550, 270) == 1)
                   setfillstyle(1, CYAN);
                   bar(310, 130, 410, 200);
                                                                         {
                   puthz(330, 150, "仓库管理", 16, 16,
                                                                              mode = 3;
BLUE);
                                                                         }
                   fillpoly(4, dian);
                   rectangle(310, 130, 410, 200);
                                                                         // Enter the past arguments
                   flag = 0;
                                                                         if (mouse_press(450, 310, 550, 370) == 1)
                   flag1 = 0;
              }
                                                                              mode = 4;
         }
                                                                             // draw_past01();
```

```
}
                                                                    setfillstyle(1,LIGHTBLUE);
                                                                    bar(585,450,625,480);
                                                                                                        跳
                                                                                                                   过
     // Enter the help arguments
                                                                    puthz(625-40+3,480-30+10,"
     if (mouse_press(450, 410, 550, 470) == 1)
                                                          ",16,16,WHITE);
    {
                                                               }
          mode = 5;
         // draw help01();
                                                               void next(void)
    }
                                                               {
                                                                    setfillstyle(1,LIGHTBLUE);
     // Enter the home page
                                                                    bar(585,450,625,480);
                                                                                                         下
                                                                                                                   页
     if (mouse_press(310, 130, 410, 200) == 1)
                                                                    puthz(625-40+3,480-30+10,"
                                                          ",16,16,WHITE);
         // draw_home01();
                                                               }
          mode = 1;
         /*else
                                                               void last(void)
                                                               {
         {
              draw_home00();
                                                                    setfillstyle(1,LIGHTBLUE);
         }*/
                                                                    bar(0,450,40,480);
    }
                                                                    puthz(4,480-30+10,"返回",16,16,WHITE);
                                                               }
     // Exit the program
     if (mouse\_press(0, 0, 40, 30) == 1)
                                                               /*void text_input(char *str,int x1,int y1,int x2,int
    {
                                                          y2,int t_x,int t_y,int t_size)
         wr_h();
                                                               {
         free(h);
                                                                    char temp,*p;
          exit(0);
                                                                    int
    }
                                                          n=t_x,get,arr[10]=\{p_0,p_1,p_2,p_4,p_5,p_6,p_7,p_8,p_9\}
}
                                                                    clrmous(MouseX,MouseY);
/*Draw the whole welcome page*/
                                                                    p=str;
void draw wel()
                                                                    setfillstyle(1,WHITE);
{
                                                                    setcolor(DARKGRAY);
     draw_wel_buttons();
                                                                    bar(x1,y1,x2,y2);
     draw_wel_cofield();
                                                                    while(bioskey(1))
     draw wel tractors();
                                                                    {
     quit();
                                                                         get=bioskey(0);
}
                                                                    }
                                                                    while(*p!='\0')
void quit(void)
                                                                    {
{
                                                                         if (get==p_Enter)
     setfillstyle(1,LIGHTBLUE);
     bar(0,0,40,30);
                                                                              break;
     puthz(3,10,"退出",16,16,WHITE);
}
                                                                         for(i=0;i<10;i++)
void skip(void)
                                                                              if(arr[i]==get)
{
                                                                              {
```

```
temp=i+'0';
                                                                033)
                   }
                                                                                   { // 033:Esc
              }
                                                                                        if (t != '\b')
              *p=temp;
                                                                                        {
                                                                                             *(id + i) = t;
              p++;
                                                                                             *(id + i + 1) = '\0';
              outtextxy(n,t_y,&temp);
              get=bioskey(0);
                                                                                             bar(SX+i*w-1+space,SY - 1,SX
                                                                +i*w+1+space,SY+h); //遮盖光标
              n+=t_size;
         }
                                                                                             if (flag)
    }*/
                                                                                                  outtextxy(SX + i * 18, SY,
                                                                id + i); //输出刚输入的字符 t
    void input text(char *id, int x, int y, int charnum, int
                                                                                             else
color, int flag)
                                                                                             {
    { // flag==1 显示
                                                                                                  outtextxy(SX + i * 18, SY,
                                                                "*");
    #define h 32
                                                                                             }
    #define w 18
                                                                                             i++;
    #define space 0
                                                                                             line(SX + i * w + space, SY, SX
    #define SX x + 5 // START X
                                                                +i*w +space,SY+h);
    #define SY y - 5
                                                                                        }
                                                                                        else if (t == '\b' \&\& i > 0)
         //int k = 0;
         int i = 0;
                                                                bar(SX+i*w-1+space,SY-1,SX+i*w+1+space,SY+h); // 遮 盖
         char t;
                                                                光标
         clrmous(MouseX, MouseY);
                                                                                             i--;
         setfillstyle(SOLID_FILL, color);
                                                                //减少一个字数
         setlinestyle(SOLID_LINE, 0, NORM_WIDTH);
                                                                                             bar(SX+i*w,SY,SX + i*w + w,
         setcolor(DARKGRAY);
                                                                SY + h);
                                                                                     //遮盖文字
         settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);
                                                                                             line(SX+i*w+space, SY, SX + i
         settextjustify(LEFT_TEXT, TOP_TEXT);
                                                                * w + space, SY + h); //绘制光标
                                                                                             *(id + i) = '\0';
                                                                                             *(id + i + 1) = '\0';
         while (bioskey(1))
                                                                                        }
         {
                                                                                   }
              t = bioskey(0);
         }
                                                                                   else
         while (*(id + i) != '\0')
                                                                                        bar(SX+i * w - 1 + space,SY - 1, SX
                                                                + i * w + 1+ space, SY+ h); //遮盖光标
              i++;
         line(SX+i*w,SY,SX+i*w,SY+h);
                                                                                        break;
         while (1)
                                                                                   }
         {
                                                                              }
                                                                              else
              setfillstyle(1,WHITE);
              t = bioskey(0);
                                                                              {
                                                                                   if (t!='\n'\&\&t!='\r'\&\&t!=033)
              if (i<charnum)
              {
                                                                                   { // 033:Esc
                   if (t!='\n'\&\&\ t!='\r'\&\&\ t !='
                                                                                        if (t == '\b' \&\& i > 0)
```

```
{
                                                                     {
                        bar(SX+i*w-1+space,SY-1,SX
                                                                         bar(SX+i*w
+ i * w + 1 + space, SY + h); //遮盖光标
                                                     -1+space,SY-1,SX+i*w+1+space,SY+h); //遮盖光标
                        i--;
                                                                         break;
//减少一个字数
                                                                     }
                        bar(SX + i * w, SY, SX + i
                                                                 }
                                    //遮盖文字
*w+w,SY+h);
                                                             }
                        line(SX + i * w + space, SY, SX
                                                             //return i;
+ i * w + space, SY + h);
                               //绘制光标
                                                         }
                        *(id + i) = '\0';
                        *(id + i + 1) = '\0';
                                                     <1>引用代码
                    }
                                                     引用来自学长的鼠标、汉字及部分写屏代码
                }
                else
```

# 八、 时间安排

第一周: 讲行需求分析并学习主要共性知识

第二周:完成需求分析并初步掌握共性知识

第三周:完成分工,建立代码远程仓库,提交需求分

析报告,开始编程

第四周: 完成欢迎界面全部内容, 完成各辅界面框架

内容

第五周: 完成全部页面基本内容, 中期验收

第六周: 优化界面设计, 优化绘图算法

第七周:继续优化算法,做抗压调试及部分改进

第八周:程序调试,整理报告,准备最后验收

## 九、工作分配及代码量

#### 梁栢杰:

注册登录及用户管理,农田农机参数的输入及保存,仓库信息的输入及保存,参数信息处理,帮助与说明界面,以及上述对应的界面设计及画图。

有效代码行数: 3301 行

#### 冯天瑞:

棉花采摘农机行进路径规划,不同形状土地提取与生成,农田参数对应产量与用时的计算,仓库管理,以及上述对应的界面设计及画图。有效代码行数: 4252 行

## 十、 组员感想

### 组员冯天瑞感想

C语言课程设计极大地锻炼了我的代码能力,并直观地教会了我现代计算机语言构建工程项目的基本方法,设计程序结构的基本思路以及项目管理思维。在完成 C课设的过程中,我不仅对上个学期的 C语言程序设计课程内容有了更加深刻的理解,也借此学习了 bc 下的图形编程和工程项目建立,并简单地学习与应用了 git 版本管理。在此过程中,有艰难 debug 的困顿,也有程序最终正常运行的喜悦,我

和队友克服一道又一道难关,从 0 开始一点一点构建作品,在多次实验中找出不足之处并加以改进,在老师和学长学姐的指导下扩展思路,设计更加人性化智能化的人机交互方式。与此同时,我在完成课设的过程中通过网络查找关于棉花机械化采摘的相关资料,收集各省不同种类棉花相关数据,尝试着构建出更加符合实际的棉花田地模式和收割机路径规划方法,并估算土地收益。

学习编程的第一步总是艰难而笨拙的,前期的 C 语言课程以及 C 语言课程设计就是这万里征途的第一步。而这第一步的回报也是极其丰厚的,由此我们体验到了程序设计的魅力,能够调整自我,敢于应对更多挑战。

最后非常感谢我的队友梁栢杰同学在文件操作上的支持与帮助, 同时也要感谢各位老师与热心的学长学姐们的意见与指导,也希望在 接下来能够继续提升自己的编程水平。

### 组员梁栢杰感想

为期近三个月的 c 语言课程设计就此落幕,在这段时间里,我受益良多。c 课设极大地提升了我的编程能力,让我从程序的黑框中跳了出来,我的程序开始有了颜色变化,也更加贴近了实际的应用。在我刚开始学编程的时候,我有想过在这个黑框中究竟能干些什么,现在,c 课设告诉了我答案。在完成 c 课设的道路上,我遇到了许多的困难,我曾因为一个不起眼的 bug 而被困扰了三天,也曾为它的功能不理想而苦恼。总的来说,从一片虚无中一步一个脚印,最终走完这

道路,可以说是我永远不会忘记的一段旅程。再者,c课设也让我接触到了 github 这个管理代码的工具,虽然我对这个工具使用并不熟练,甚至弄丢了两次代码,但我仍然认为他是一个多人合作项目必不可少的工具。

整个任务做下来,我认为最重要的就是耐心,细致,和坚韧的心态。无论是找 bug,还是写代码,有了这些东西,往往就会事半功倍。除此之外,我认为最重要的就是经验,整个 c 课设的过程也是积累经验的过程,一个问题,你遇到过,并解决了,那他对你来说就不是问题,但对其他人来说这或许就能要了他们的命。同时,学长所提供的各种教材和祖传代码也是我们完成 c 课设的关键。

最后,非常感谢在 c 课设过程中帮助过我的老师,助教,学长和同学,他们或多或少都对我脆弱的心灵施以了援手,为我做 c 课设增加了动力。我还要感谢我的队友冯天瑞,他让我可以完全专注于我自己的任务,并在我需要帮助时对我提供帮助。

# 十一、参考文献

- 1. 王士元. C 高级实用程序设计. 北京: 清华大学出版社. 1996
- 2. 周纯杰, 刘正林等. 标准 C 语言程序及应用. 武汉: 华中科技大学出版社. 2008
- 3. 周纯杰,何顶新等.程序设计教程 用 c/c++语言编程.北京: 机械工业出版社.2016