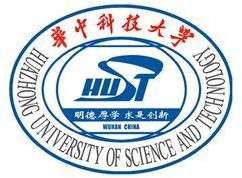
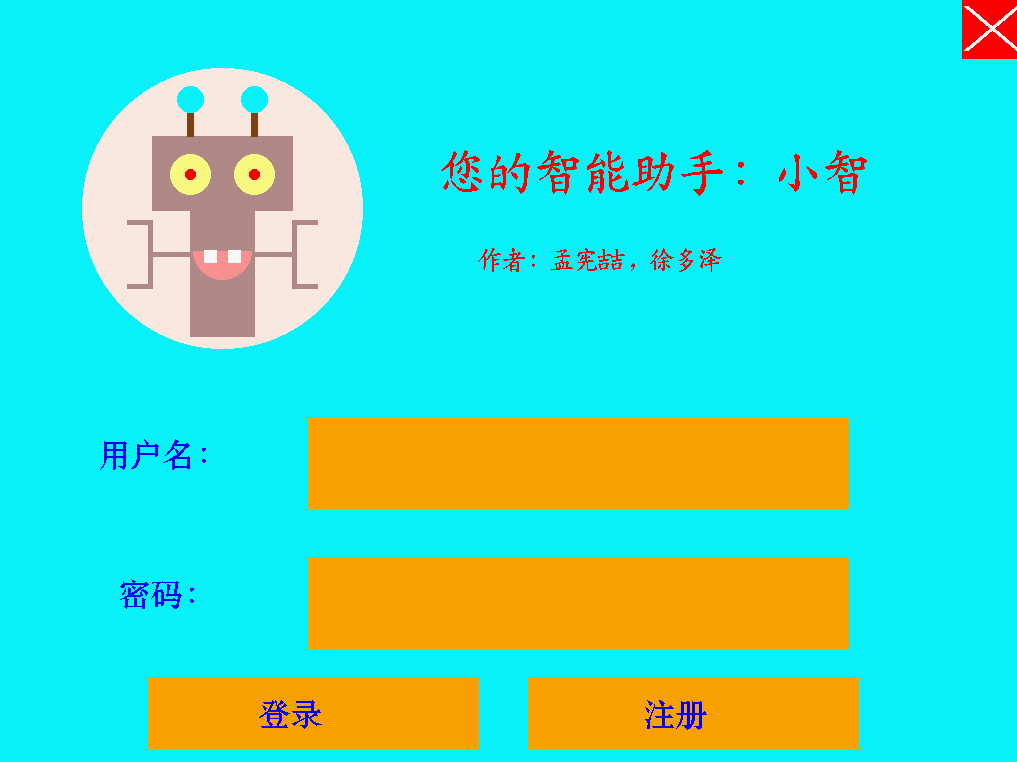
****

**华中科技大学**

**C语言课程设计报告**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **题目：** | **生成式AI** | | |
| **班级：** | **自动化创新实验班2401班** | | |
| **组员：** | **孟宪喆** | **学号：** | **U202410203** |
| **组员：** | **徐多泽** | **学号：** | **U202414181** |
| **指导教师：** | **周纯杰、何顶新、彭刚、周凯波、桑农、左峥嵘、高常鑫、汪国有、陈忠** | | |
| **上交日期：** | **2024-4-27** | | |

目录

[一、前言 3](#_Toc12955)

[1. 选题介绍 3](#_Toc4144)

[2. 编写背景 3](#_Toc12183)

[3. 编写目的 3](#_Toc11935)

[4. 参考资料 4](#_Toc18620)

[二、任务概述 4](#_Toc26619)

[1. 需求分析 4](#_Toc26602)

[2. 可行性分析 5](#_Toc7114)

[3. 编写规范 5](#_Toc16145)

[三、 运行环境和配置 6](#_Toc19129)

[1. 硬件接口 6](#_Toc28862)

[2. 软件接口 6](#_Toc26470)

[3. 控制 6](#_Toc15925)

[四、 程序设计 7](#_Toc27302)

[1. 主要功能 7](#_Toc28461)

[2. 创新点 7](#_Toc11957)

[五、 程序设计与算法实现 8](#_Toc2225)

[1. 综述 8](#_Toc28983)

[2. 协议栈 9](#_Toc21417)

[1） 数据结构 9](#_Toc6643)

[2）分词标号协议 11](#_Toc31007)

[3）语言交换协议 11](#_Toc5293)

[4）语料查询协议 17](#_Toc26942)

[3. 文生文算法 17](#_Toc16012)

[1）第一步：分词 17](#_Toc11152)

[2）过渡步：初步标号 18](#_Toc11243)

[3）第二步：抽象标号 18](#_Toc27159)

[4）第三步：逻辑查找+实现回答 19](#_Toc29068)

[5）隐藏步：其他隐藏的步骤 21](#_Toc22060)

[4. 计算 22](#_Toc13444)

[5. 文生图 23](#_Toc15684)

[六、 源代码以及相关文件 24](#_Toc7767)

[1. data文件 24](#_Toc5400)

[2. include头文件<.h> 24](#_Toc32537)

[3. source函数<.c> 57](#_Toc16827)

[4. 互信息值训练函数 782](#_Toc30106)

[5. stceAC.h stce.cpp 792](#_Toc11677)

[七、 课设感想 814](#_Toc30178)

[八、 课设分工 816](#_Toc19100)

# 一、前言

# 1. 选题介绍

**生成式AI模拟系统：**模拟生成式大模型的主要功能，包括自动问答、文生图等功能，重点关注功能和应用场景。

# 编写背景

人工智能迅猛发展的当下，生成式AI逐渐成为人们工作和生活的主流，于是，对AI的理论算法进行各种研究，如何使AI更快地运行，如何提高输出的准确率，如何对AI生成算法进行实质上的创新，甚至是研究如何将生成式AI迁移到Borland C++当中，对于社会具有十分重要的意义。生成式AI功能繁多，我们选择了文生文、文生图两个主要功能展开构建，并提出了新的数据结构、算法样式和NLP语义分析方法，这对于AI的理论创新也有着较大的意义。

在项目设计的过程中，我们了解到两种AI，Retrieval-based AI和Generative AI，在充分了解二者的优缺点之后，我们决定进行Generative AI的搭建。考虑到Borland C++内存的限制，以及其不能联网爬虫、GPU算力支持度不够等因素，我们以连接主义学派的理论作为依据，结合符号主义学派的研究成果，对数据结构进行了创新，提出了一系列新的思路。

# 编写目的

我们编写了这份报告，以进行宏观把控指导。该项报告对生成式AI的架构进行了理论分析、数学论证、需求和功能分析以及成果展现，并给出了该成果对应的代码和数据库。此外，本报告明确了本软件系统架构设计、软件结构设计、各模块之间的接口调用、系统界

面、系统功能设计等内容。

在报告正文的最后，我们明确了两位开发者的设计和分工，增强了后期测试人员对于软件的调试和验收的可读性与可修改性。

本报告的预期受众为：尝试寻找LLM之外的AI新模型的科研人员；连接主义学派学者；对AI新功能感兴趣，又有一定计算机基础的大众群体。

# 参考资料

1. 杨将新.C语言开发全程指南.电子工业出版社
2. 周纯杰 何顶新 周凯波 彭刚 等.程序设计教程 用C/C++语言编程.北京：机械工业出版社
3. Aston Zhang,Zachary C.Lipton,Mu Li,Alexander J.Smola.动手学深度学习.人民邮电出版社
4. 刘杰飞.自己动手做聊天机器人.中国水利水电出版社
5. 中国科学院 宗庆后研究员 研究生课程：自然语言处理
6. Thomas H.Cormen,Charles E.Leiserson,Ronald L.Rivest,Clifford Stein.算法导论（第三版）.机械工业出版社
7. 其他相关参考材料，来自csdn博客和百度飞桨公开课。

# 二、任务概述

**1. 需求分析**

·用户可以与AI进行基本的闲聊交互，当用户对AI的内容提出质疑时，AI可以善意道歉。

**·**AI可以理解用户的语言，也可以对语言做出回应。用户可以通过向AI提问的方式，获取知识性的信息。

**·**AI可以对用户给出的简单算式进行运算，保证用户采用不同的表达都能正常运算，并且保证计算结果不受数据精度影响，如果出现除以0的情况能及时给出报错。

·AI可以进行简单的数论等方面的数学运算。

**·**AI可以实现文生图功能，对于用户的生成图片需求，AI可以生成相应的内容。

**·**AI支持对历史记录进行查询，当聊天次数过多时，AI自动开启新对话。

1. **可行性分析**

·文生文的相关可行性理论证明见第五章节第3小节。

·文生图的相关可行性理论证明见第五章节第4小节。

·可通过在对话框的输入，将内容传送给AI，AI处理后给出相应的反馈。

·关于敏感词语的查找，我们曾经尝试使用AC自动机，该自动机采用有8层节点的128叉树进行BFS查找。然而，最终程序运行失败，我们将其归因于编译器的内存不足。于是，我们放弃了AC自动机算法，改用简单查找算法，这标志着算法完全具备了可行性。

1. **编写规范**

**·命名规范**

1. 函数命名采用下划线命名法以表明相关功能。
2. 文件命名能够清晰表达出所包含函数的主要功能。
3. 涉及数据结构的命名参考数据结构（C语言版），并进行适当修改。

**·注释**

1. 函数功能都要在函数原型之前注明。
2. 部分令测试者比较难以理解的算法和流程应该给出相应的注释。
3. **运行环境和配置**
4. **硬件接口**

处理器：Intel 奔腾4及以上。

硬盘：空间500MB以上。

屏幕适配器：SVGA接口。

系统运行内存：要求32MB以上。

1. **软件接口**

开发软件工具：Borland C 3.1。

操作系统：DOS WINDOWS XP。

文字编辑工具：Dev C++ , Visual Studio 2019 Community

1. **控制**

**·该系统通过鼠标与键盘实现对界面与功能的控制。**

在进入与注册界面，通过鼠标点击相应的文本框或按键，进入相应功能，点按ENTER键退出输入功能。

在AI交互界面，通过点按按键切换文生文与文生图功能与发送文本，通过在文本框内写入汉字，实现文本的输入。

在输入框内，键盘点按ENTER键可以实现输入内容的清除；考虑到汉语语法与英语语法的差异性，我们禁止了英语字符的输入，但数字、运算符、标点可以正常输入。

1. **程序设计**
2. **主要功能**

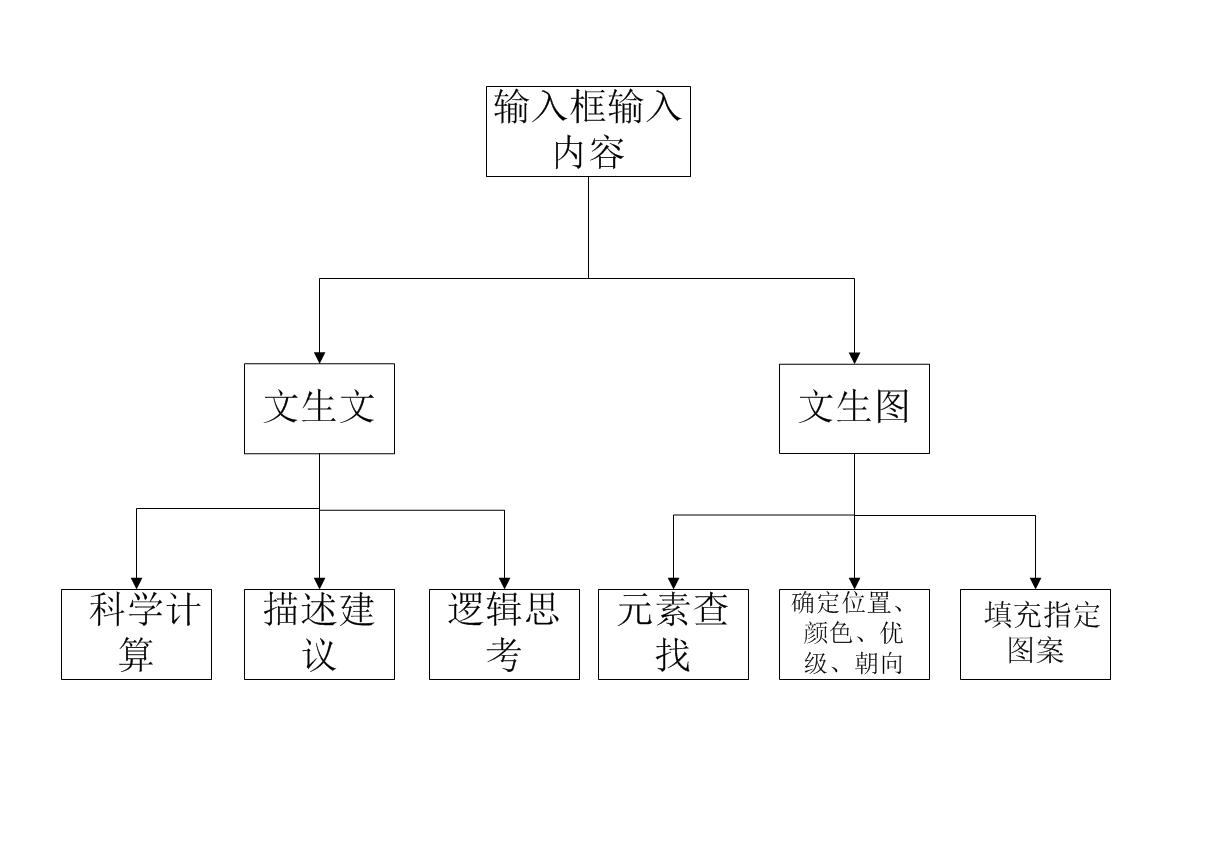
1）实现注册/登录。

2）文生文，提供知识性信息的问答。

3）文生图，根据用户需求对图像进行绘制。

4）对四则运算、最大公约数、判断是否为质数等内容进行数学分 析并给出结果。

5）实现点按上下键翻看历史记录，并在聊天达到一定次数之后自动 开启新对话。



**2. 创新点**

**我们实现了诸多A院首例：**

1. 从画点函数开始，通过Bresenham算法实现简单图形绘制，通过 搭建相关的鼠标函数硬件接口，通过自己绘制ASC2码字符，搭建 了A院历史上第一个SVGA 1:5:5:5的65536色界面，为单独的透 明度架构搭建了窗口。
2. 实现了A院第一个SVGA输入法，解决了SVGA不能输入汉字的问 题。
3. 搭建了Bezier曲线，使得连点描曲线在A院C课设历史上第一次 成为可能，解决了复杂图形绘制只能依靠大量直线叠加的代码 冗余问题。
4. 架构内部填充函数，实现了A院历史上第一个从内部填充不规则 闭合曲线的算法，解决填充内部造成的大量代码冗余与工作量， 使得画大型不规则图案成为可能。
5. 互信息值的使用，使得信息论第一次被引入C课设；同时，通过 20小时的数据训练，使得C课设当中第一次出现了“提前跑数据， 之后用”的情况。
6. 我们在C课设当中第一次架构了大规模的协议栈，从而实现自然 语言与机器语言的相互转化以及机器语言的处理，这在A院也是 首例。
7. **程序设计与算法实现**
8. **综述**

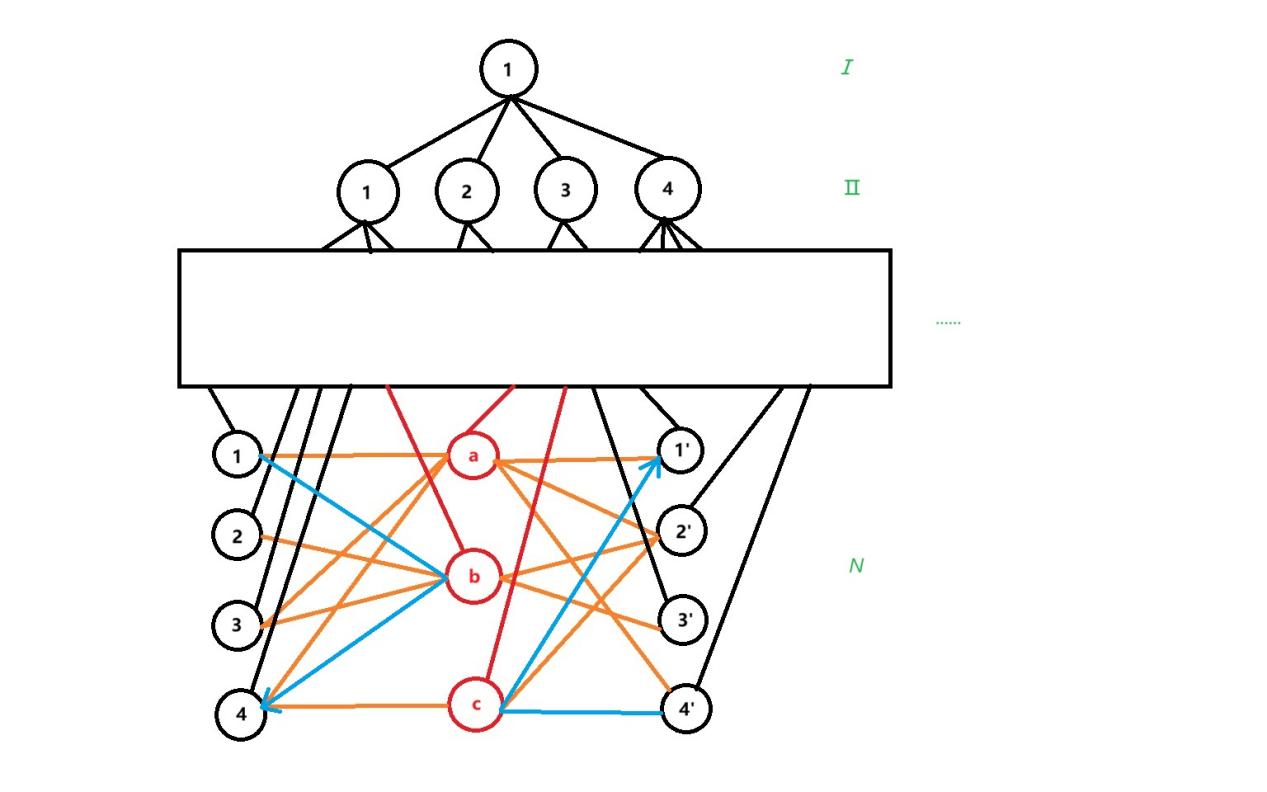
文生文功能的算法，我们将其拆成以下几个步骤：

1. 预先步：检测是否需要进行数学运算，如需要，跳转数学运算功 能。
2. 特殊步：对于特殊高频率表达直接跳转对应回答，优化时间复杂 度。
3. 第一步：利用互信息值训练出的数据库，对输入的字符进行分词 处理。
4. 过渡步：利用分词标号协议，实现对句子的标号。
5. 第二步：利用语言交换协议，通过对句子的语义和语法分析，将 自然语言彻底抽象为机器语言。
6. 第三步：利用语料查询协议，通过对知识库inf的查询，实现内容 的搜索与输出。
7. 隐藏步：查询敏感词；查询是否有重复对话；最终输出的扫尾操 作。

文生图功能的算法，采用优先级策略，即定义优先级，优先级低者作为优先级高者的背景。我们预先画出了一部分图案，这些图案也反映了我们Bezier曲线和内部填充函数的优越性。随后，根据用户输入的语言，我们将图案显示在画框中。

1. **协议栈**
2. **数据结构**

我们定义了一个有向图作为本算法的数据结构。该数据结构是本协议栈的理论基础。



上图显示的是我们创造的词典树的改良版数据结构。在该数据结构中，父结点与子结点是继承关系，即子结点继承父结点的全部性质；而对于同级的有向网络来说，我们发现了以下事实：

**·Assumption:在自然语言的表述中，所有的非感叹性陈述性表达，可以归为两类：给出主体和客体的双目逻辑关系，其逻辑连接以谓语的形式出现；给出主体单独的状态逻辑表征。**

基于该假设，我们给出了同层级之间的有向网络。左侧黑色节点表示主体，通常以主语的形式出现；右侧黑色节点表示客体，通常以宾语的形式出现。对于中间的红色节点，其表示的是主体单独的状态或主体与客体的逻辑连接。

基于此数据结构，我们给出了**熔断性继承**的概念。这是我们关于人类思维的一个猜想。如果不考虑论文《Attention is all you need》的观点，当我们对一个词语进行联想时，首先对同层级的词语进行联想（1）；当对于同层级联想不成功，则追溯到上一级进行联想（2）。此时，该词语具备了上一级的性质，同一级的查询被熔断。对于上一级的性质，我们继续联想，如果联想依然不成功，我们有两种选择：第一是向上一级联想，第二是向下一级其他词语联想（3）。在此过程中，我们采用注意力判断进行理论修正，得出下一步应该进行查询的内容，随后对上一个查询的节点进行熔断。

我们注意到，我们已经搭建了人类思维的基本范式。（1）是逻辑推理，（2）是归纳推理，（3）是类比推理，而这些的组合，更是天马星空的猜想和脑洞。如果允许知识库根据聊天内容进行更改，那么也许聊天内容会极其接近于人类。

然而，该理论仍存在不足，需要引入“自检点”的概念进行修正。在知识结点当中，散布“自检点”，对于自身思考时间和当前思考结点与问题结点的距离进行检验，从而避免思考不出来陷入死循环或给出毫不相关的答案。

**2）分词标号协议**

我们首先对词语进行了词性分类。按照语言学惯例，我们将其分为8类：名词、动词、形容词、副词、连词、介词、代词、数量词。随后，我们架构了有三级结点的字典树，对每一级进行了标号。标号的格式为，“a.b.c.”，其中c后面的点不可缺省。越往前的代表的结点越接近树根。在文件当中，其存储方式为：a.xx词 1.#类型 1.1 词 ... 2#类型 2.1 词 ...

关于字符串，我们规定分词后的字符串，其分隔用||表示；标号后的字符串，其相当于词语标号的从前向后的拼接。

**3）语言交换协议**

**本协议的目的是将自然语言转换为机器语言。**

汉语语素的构成：

主语、谓语、宾语、定语、状语、补语、标志语 ->提出标志语的概念。

下面对每个类型的词语进行语法标记。

谓语：

第1位是时态，0是现在，1是过去，2是未来

第2位是已然与未然，0是在做，1是完成，2是要去做

第3位是语态，0是主动，1是被动

第4位是所带状语的个数，0是完全没带状语

第5位是所带补语的个数，0是完全没带补语

第6位是逻辑关系

0：主语=宾语 1：主语>宾语 2：主语<宾语

第7位是词语词性，0是动词，1是形容词

第8位是词语分类

第9位是分类对应的编号

第10位是分类对应的三级编号

第11位开始，是状语；然后是补语

主语、宾语：

第1位是语气，0是陈述语气，1是疑问语气，2是感叹语气

第2位是与谓语的逻辑关系

0：被施予 1：施予

第3位是与另一类的逻辑关系

0：主语=宾语 1：主语>宾语 2：主语<宾语

第4位是and,or问题：0是一般，1是and，2是or

第5位是and,or有几个：要是第4位填0，第5位也填0

第6位是定语个数

第7位是词语分类

第8位是分类对应的编号

第9位是分类对应的三级编号

第10位开始，是并列短语，然后是定语

定语：

第1位是情感倾向，0是平淡，1是愤怒，2是惊讶，3是沮丧，4是快乐，5是恐惧，6是悲伤

第2位是与修饰词的逻辑关系

0：定语=修饰词 1：定语>修饰词 2：定语<修饰词

第3位是与其他定语的逻辑关系，0是没有其他定语

第4位是词语分类

第5位是分类对应的编号

第6位是分类对应的三级编号

状语：

第1位是情感倾向，0是平淡，1是愤怒，2是惊讶，3是沮丧，4是快乐，5是恐惧，6是悲伤

第2位是与修饰词的逻辑关系

0：状语=修饰词 1：状语>修饰词 2：状语<修饰词

第3位是与其他状语的逻辑关系，0是没有其他状语

第4位是词语分类

第5位是分类对应的编号

第6位是分类对应的三级编号

补语：

第1位是情感倾向，0是平淡，1是愤怒，2是惊讶，3是沮丧，4是快乐，5是恐惧，6是悲伤

第2位是与修饰词的逻辑关系

0：补语=修饰词 1：补语>修饰词 2：补语<修饰词

第3位是与其他状语的逻辑关系，0是没有其他状语

第4位是词语分类

第5位是分类对应的编号

第6位是分类对应的三级编号

标志语：

用来调整其他词语的前置代码

句子的标记：

首先是句子的起始部分，255标志开始

然后是主谓宾定状补，如果没有就是255代替

接着是句间关系，如果没有句间关系就是255，否则：

第一是关联句子数；

第二，交错出现：句子标号与逻辑关系

1并列关系、2承接关系、3递进关系、4选择关系、5转折关系、6假设关系、7因果关系、8条件关系、9解说关系、10目的关系

最后是句子的标号，从1开始记录

关于情感的部分，我们借鉴了上世纪情感分类学的研究主流成果，将情感分成五类，但是考虑到语言没有情感的可能性，我们多给出一个平淡情感。这就形成了上面的格局。

在存储时，将上述内容存入ZHUWEIBIN结构体中，其定义如下：

struct ZHUWEIBIN

{

int zhu[30];

int wei[30];

int bin[30];

};

数组zhu负责存储主体，wei负责存储逻辑关系，bin负责存储客体。

其对应标志词如下：

**关于时态：**

**过去：**曾、曾经、已、已经、了、过、刚、刚才、方才、刚刚、先、先前、之前、往日、往昔、从前、过去、早年、早年里、早前、早年间、昔日、曩昔、隔夜、头天、上回、上次、那会儿、那时、当时、彼时、昨天、前天、几天前、去年、前年、大前年、大前天、昨晚、昨天上午、上周、几周前、几年前、几个月前、几十年前、上世纪、二十世纪、初、民国时期、民国初年、去年同期、很久以前、在此以前、在那以前、追溯到、追忆往昔、往、往昔、曩、曩者、前番、前儿、前儿个、头晌、头晌儿、老早、老早以前、前些年、数年前、旧年、旧时光、往时、往岁、往载、曩日、曩岁、曩代、往古、畴昔、素昔、向者、尝、几天之前、几个月之前、几年之前、上个月、上月、当年、彼时、彼时彼刻、那阵儿、那些年、当年、当年间、曾经一度、曾几何时、忆往昔、遥想当年、过往、往昔岁月、过往时光、过去的日子、往日时光、过去之际、之前那段时间、以往、往日里、昔日时光、前些日子、往年初、去岁、早年期间、昔日岁月、往日光阴、曾经那段时间、早年头、数载之前、往昔之际、过去那段日子、旧时、往月、去月、往昔年华、往昔流年、很久以前、不久前、前些年、那年、那天、那时节、过去的、前儿、前儿个、上上周、上上个月、上季度、前番

**现在：**现在、此刻、此时、当下、眼下、如今、今朝、当今、现而今、这会儿、这会子、正当下、当下之时、目前、目下、方今、现况下、现今、如今、这会儿、这会子、即时、即刻、随即、当下之间、现时代、现世、现时段、当下时刻、此刻时光、眼下光景、今朝岁月、当今之际、现今之时、如今之日、正这会儿、当下这会儿、此时此刻、当下之际、当下之时、目今、现目前、当下这一瞬、当下这一时刻、当下这一时段、就在此刻、就在当下、现刻、现今当下、现期、现时代当下、这个时代、在这个时代

**未来：**未来、将来、明日、明日之后、来日、来日方长、日后、往后、往后日子、今后、未来时光、未来岁月、未来日子、未来之际、未来之时、未来阶段、以后、此后、长远来看、远期、远景、展望未来、若干年后、迟早、早晚、总有一天、明儿、来日、日后、将来、未来、未来岁月、未来时光、今后、往后、往后日子、往后岁月、往后时光、以后、此后、迟早、早晚、终有一日、总有一天、未来某时、未来某天、未来阶段、未来时期、远期、远景、远景规划、展望未来、若干年后、许久之后、过些时日、等到那时、从今往后、从今以后、从今起、自今而后、后天、明天、下周、下个月、过几天、到时候、下下周、几天后、几周后、几个月后、几年后、明年、大后天、大大后天、后年、大后年、大大后年、翌日、明晨、明晚、未来时光、未来日子、未来阶段、未来岁月、未来世代、今后岁月、往后日子、往后时光、远期、远景、来日、来日可期、未来可期、明日之星、未来之星、明日黄花

**关于已然与未然：**

**在做：**着、呢、正、正在、现在、当下、眼下、之际、这会、在这时、这时

**完成：**了、过、已经、刚刚、了呢、来着、业已、已然、曾、曾经、刚、才、早已、都、方、方才、适才、毕、罢、完、成、终于、终究、算是、讫、之后、以来、就、便

**要去做：**马上、立刻、即将、就要、准备、打算、计划、正欲、刚要、将要、快要、要、想、去、眼看、行将、决意、立意、有心、着手、动手、开手、欲、拟

**4）语料查询协议**

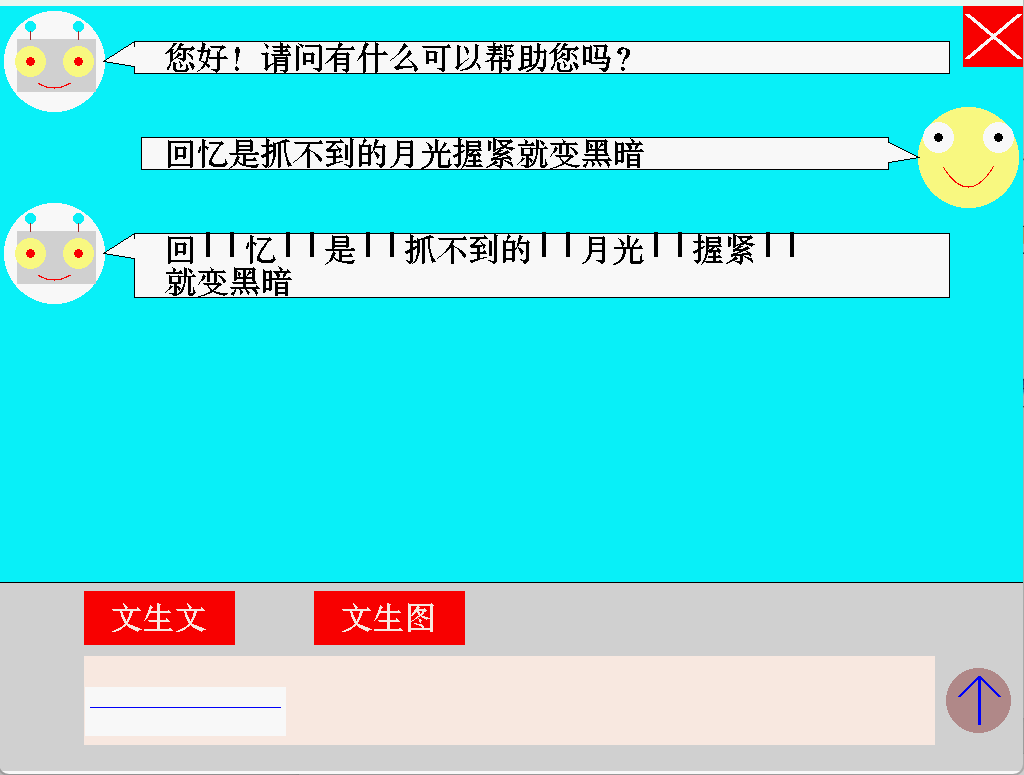
在文件当中的语料如此保存：首先是词语本身，然后用$985分隔开，下面的词语的定义；然后用$985分隔开，再往下是相关的逻辑关系。每一组字符中，-=后是谓语逻辑，+=后是宾语客体。语料查询时，即根据此方式，科学查找。

1. **文生文算法**

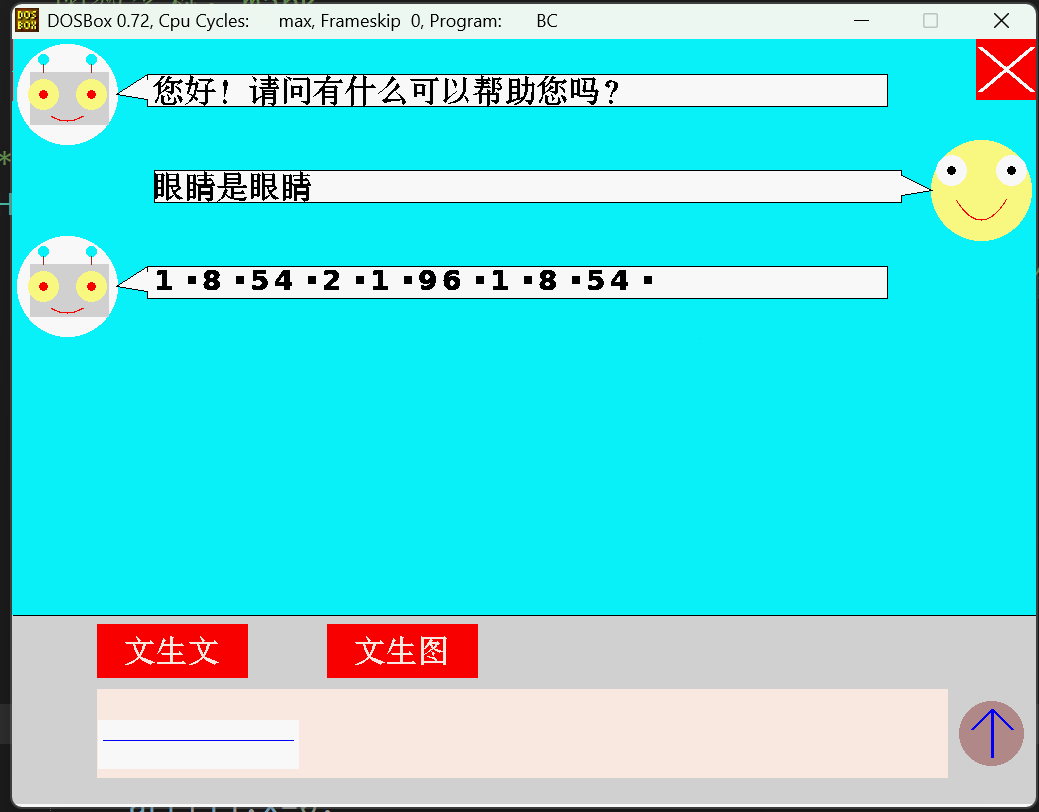
**1）第一步：分词**

我们在知乎上寻找了16万字，在弱智吧寻找了2万字修正，最后采用训练函数实现了分词数据库的训练。

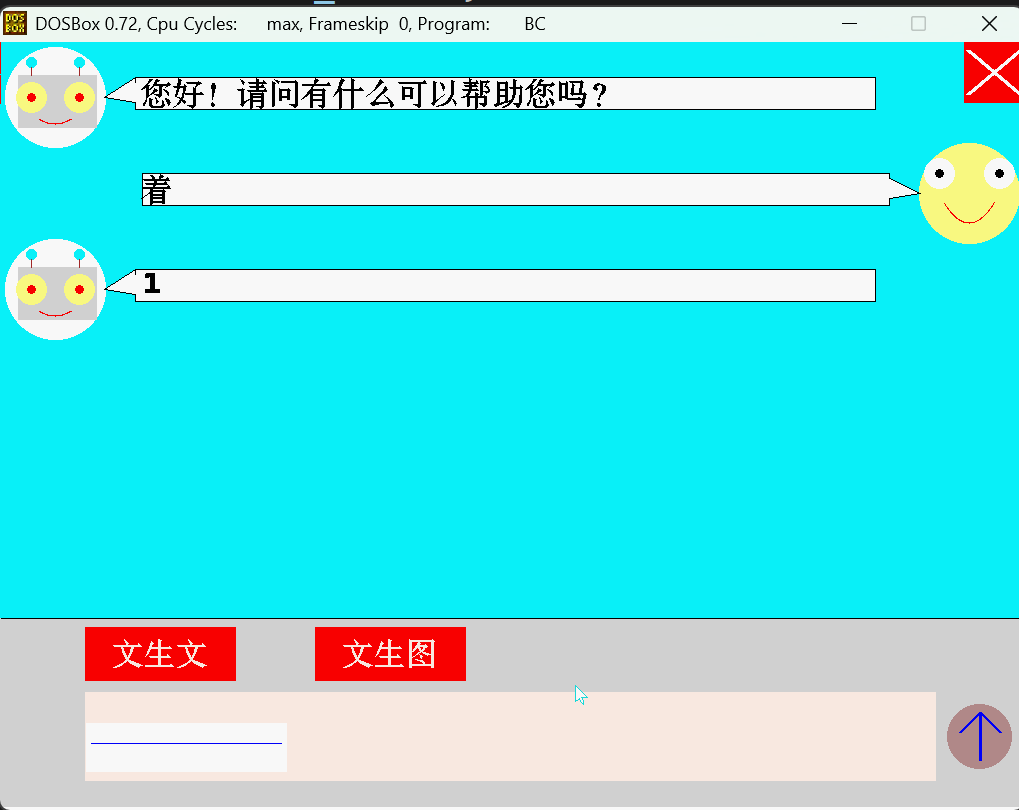
在fenci.cpp中，我们成功实现了分词。运行如下：



**2）过渡步：初步标号**

在trans12.cpp中，我们成功实现了初步标号。运行如下：

**3）第二步：抽象标号**

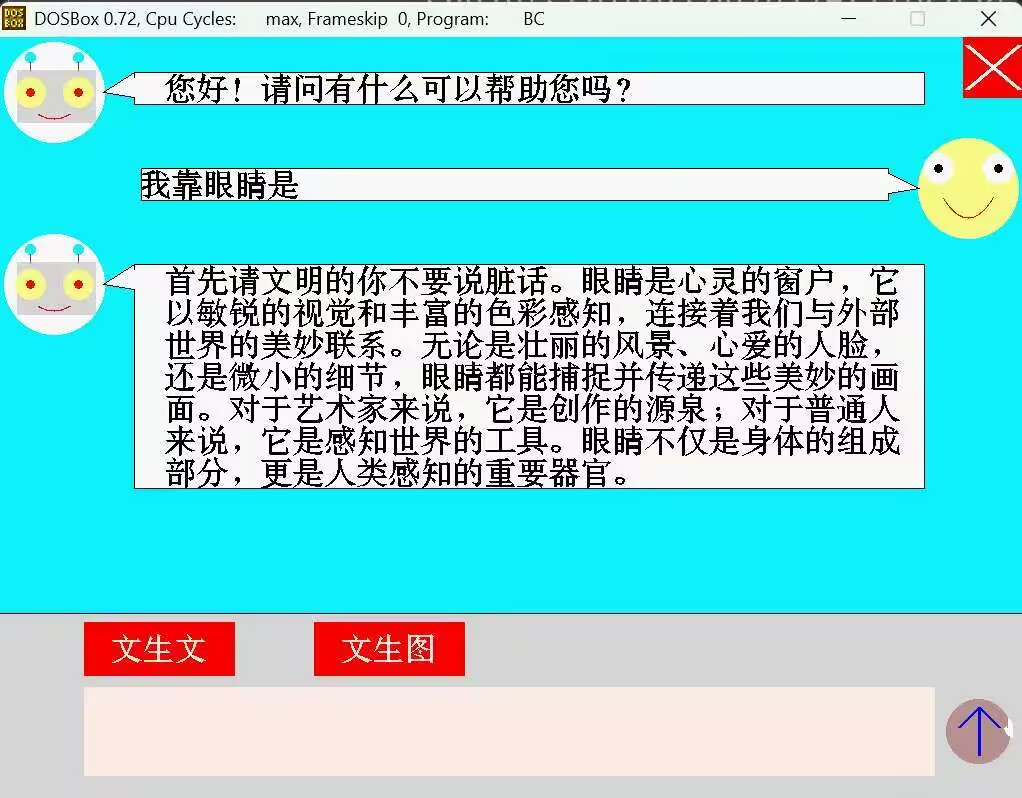
在lasthao.cpp和stce.cpp里面，我们实现了抽象标号。具体运行效果如下：

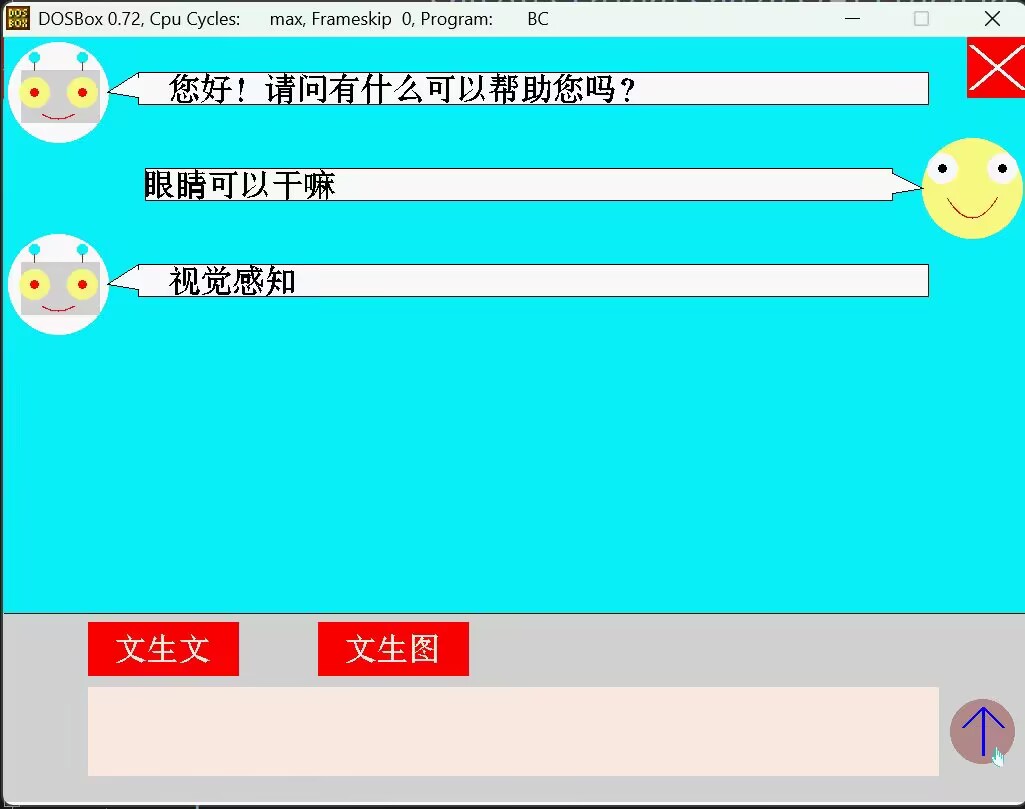
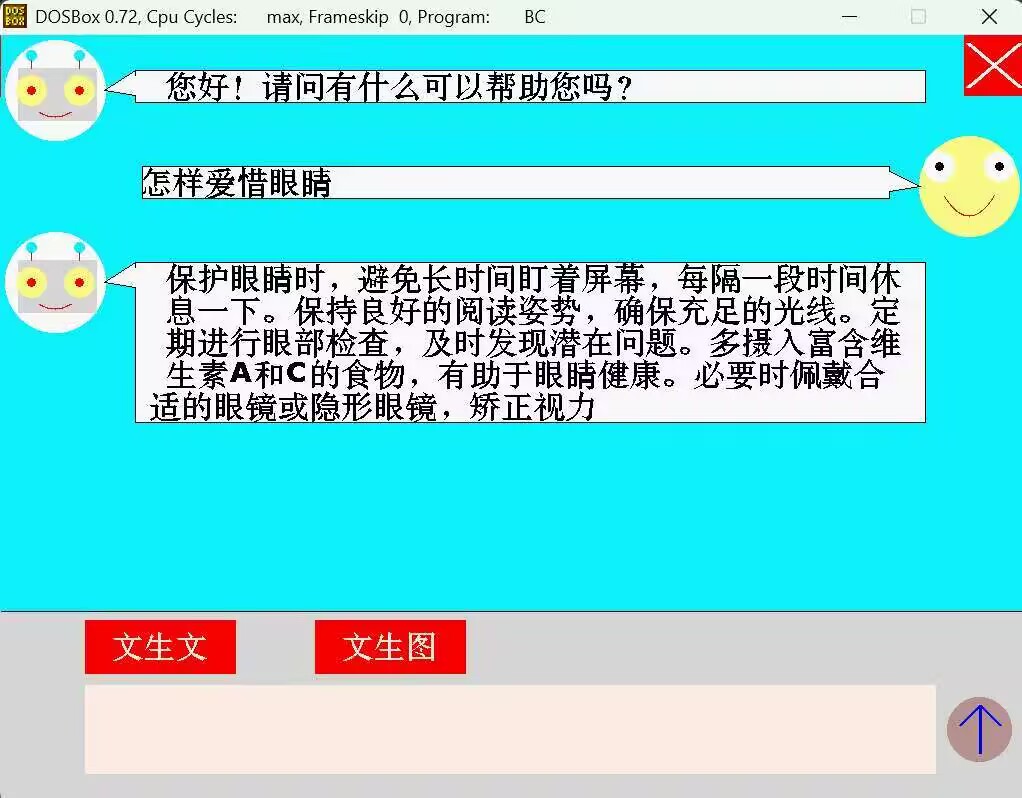
该图反映了实现确定已然与未然的功能。

另外，我们敲了stceAC.cpp，以期实现AC自动机的查找。然而，囿于Borland C++的内存有限，我们的AC自动机BFS查找算法没有顺利实现，然而，这也为查找提供了一个新的思路。

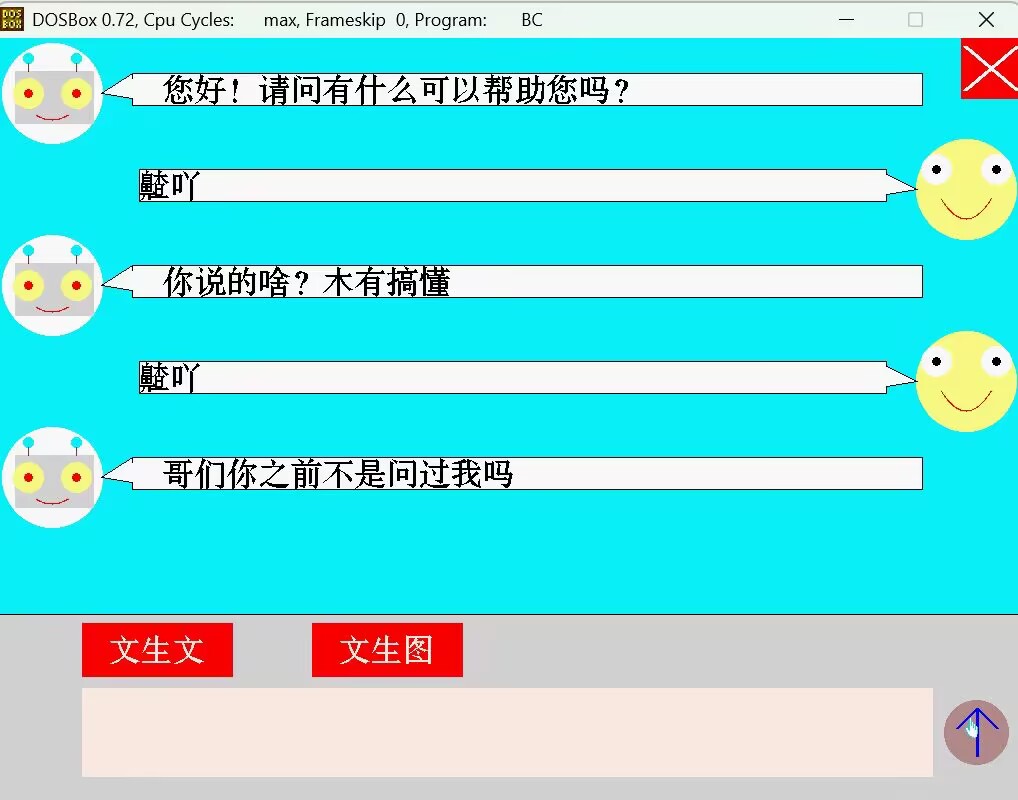
1. **第三步：逻辑查找+实现回答**

这一步全部集中在find.cpp当中，是生成文本的最后一步。实现如下：



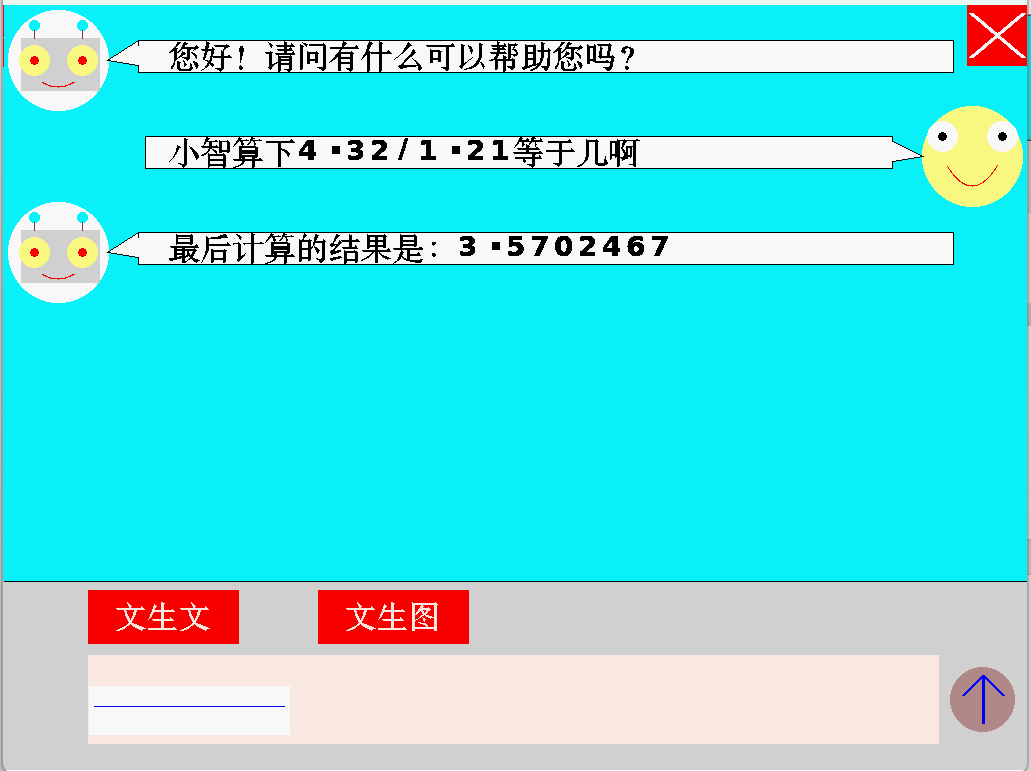
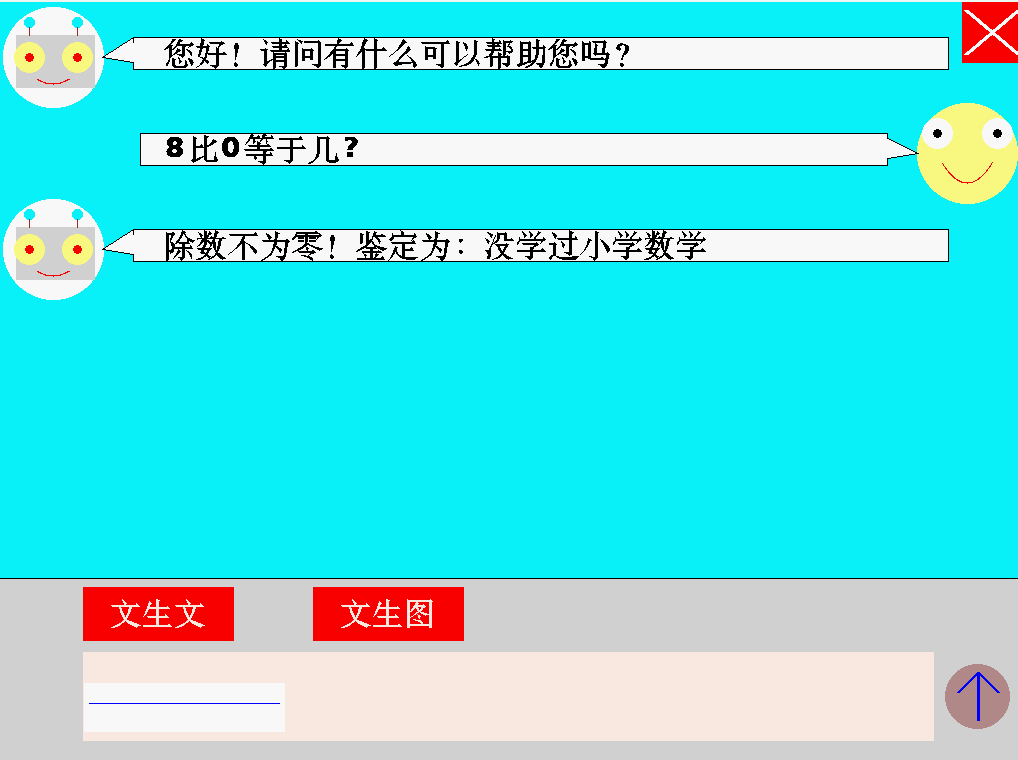


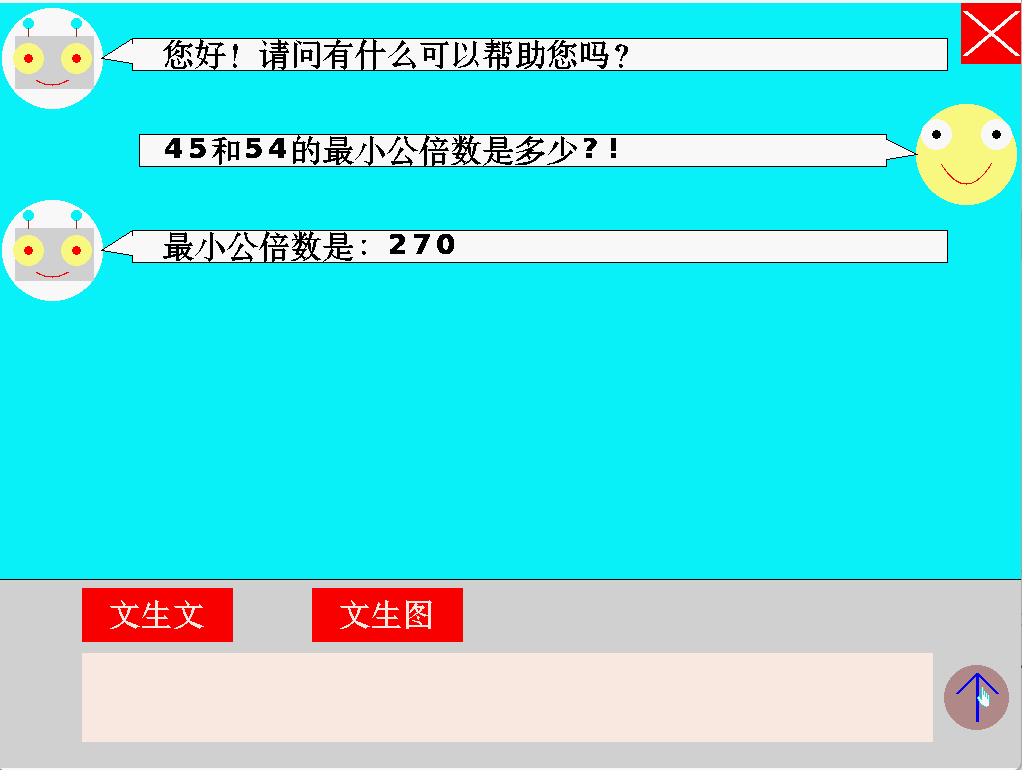
**5）隐藏步：其他隐藏的步骤**

****

1. **计算**

我们的计算采用高精度算法，支持用户用千奇百怪的方式输入，并计算出相应的结果。不用担心数据溢出。



1. **文生图**

我们采取的策略是，先画出组成整体的元素，再给出这些元素的优先级，然后进行优先级掩盖的画图，优先级低的被优先级高的部分掩盖。例如，山可以作为背景，因此可以作为低优先级出现，从而被其他物体掩盖。我们的画图采用了Bezier曲线和内部填充函数，这充分证明了我们画图的优越性。

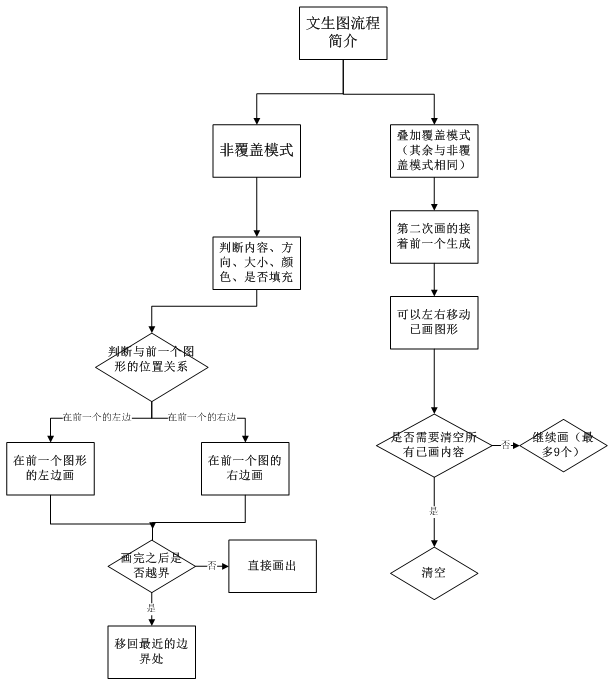
为了进行图像元素之间的交互，我们：

·实现了图像填充与否的判断。

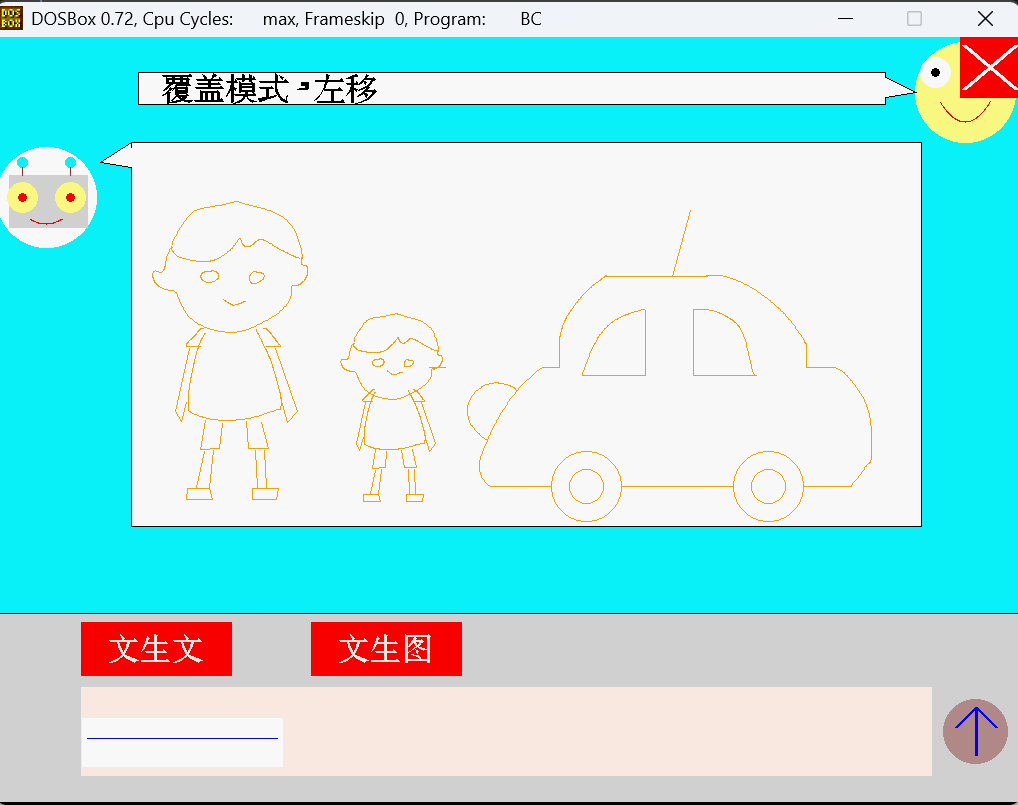
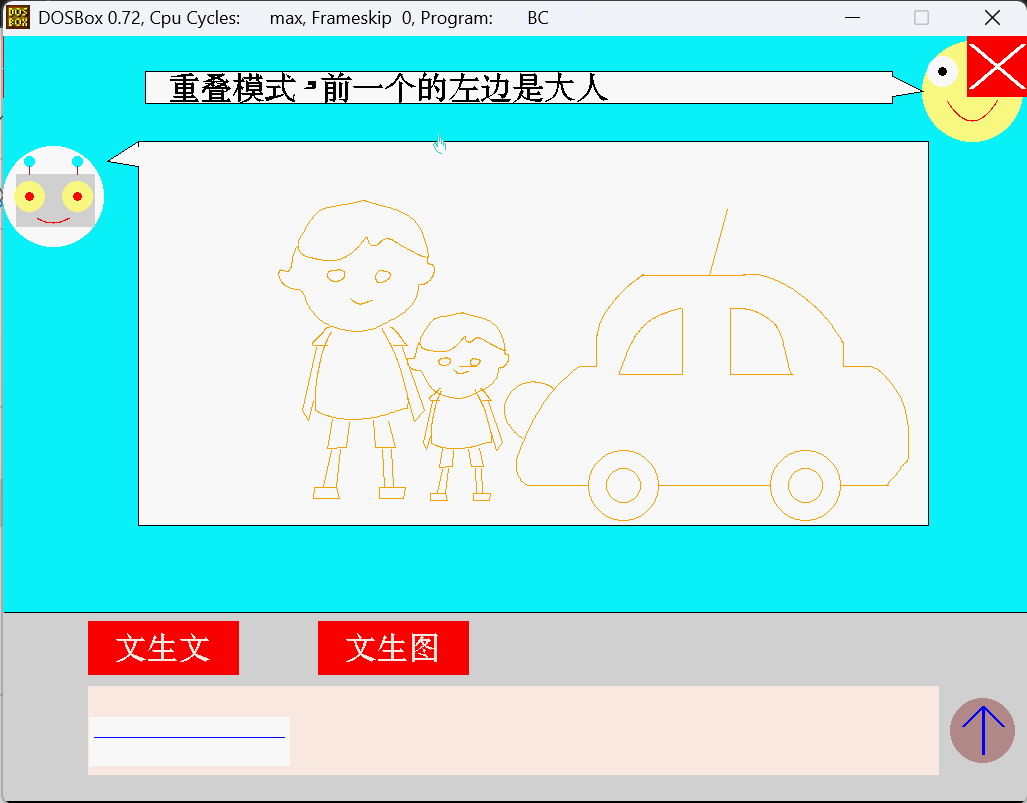
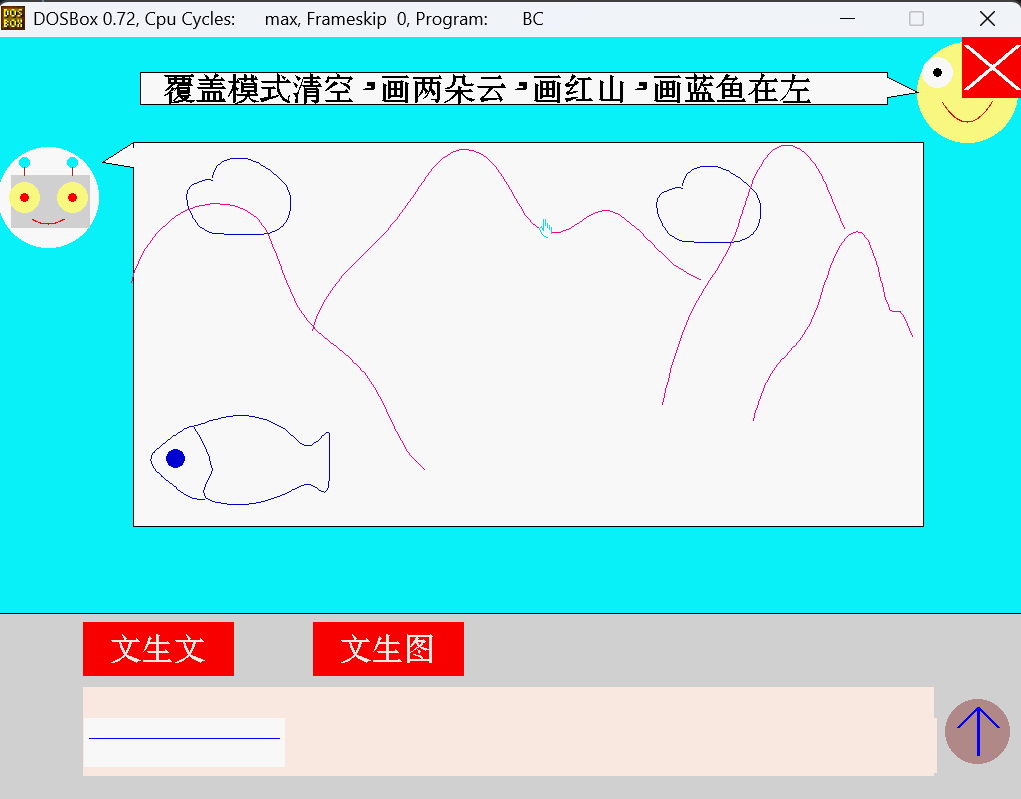
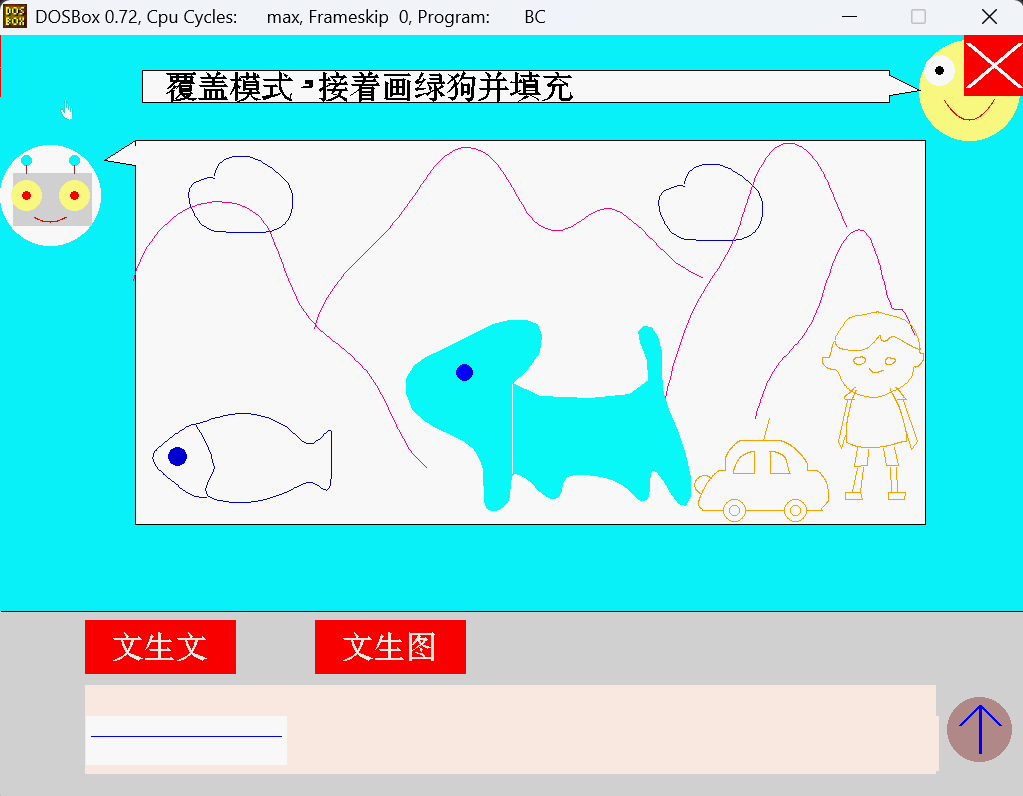
·实现了图像颜色和大小的判断。

·实现了图像绝对位置（相对文本框）、相对位置（相对其他图像） 和朝向的绘画。

·实现了图像个数的绘制判断。



下图给出文生图的几个实例。



**六、源代码以及相关文件**

1. **data文件**

**屏幕截图 2025-04-23 014254**

1. **include头文件<.h>**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_42km\_h\_

#define \_42km\_h\_

#define ADD 1

#define SUB 2

#define MUL 3

#define DIV 4

#define PRIME 5

#define ZUIXIAOGONGBEI 6

#define ZUIDAGONGYUE 7

#define SQRT 8

int calculate\_all(char\* input,int sum,char\* input2);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_AI\_h

#define \_AI\_h

#define ROBOT\_ROLE 0

#define USER\_ROLE 1

void entrance(int page,int\* keyboard,int status,char input[51][421],int\* input\_counter);

int text\_height(char input[51][421]);

int text\_jushu(char input[51][421]);

int text\_position(int height,int pianyi);

int display\_text(char input[51][421],int pianyi,int page,int keyboard);

int pianyiliang(int sum\_height,int height,int keyboard);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_Bezier\_h\_

#define \_Bezier\_h\_

struct Vec2

{

double x, y;

};

void ellipse\_(int centerx, int centery, int a, int b, int color, int page, int offsetx, int offsety);

int get\_n(int color, int\* k, int i, int page);

void fill\_1(int color, int page);

void fill(int x, int y, int color, int page, int status);

void Bezier(int color, const unsigned int length, int page, int\* x, int\* y, int status);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_boy\_h\_

#define \_boy\_h\_

void boy\_hair(int x,int y,float k,int color,int page);

void boy\_face(int x,int y,float k,int color,int page);

void boy\_eye(int x,int y,float k,int color,int page);

void boy\_body(int x,int y,float k,int color,int page);

void boy\_leg(int x,int y,float k,int color,int page);

void boy\_arm(int x,int y,float k,int color,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_car\_h\_

#define \_car\_h\_

void car(int x,int y,float k,int color,int page);

void car\_light(int x,int y,float k,int color,int page);

void car\_windows(int x,int y,float k,int color,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_color\_h\_

#define \_color\_h\_

#define SPRING\_GREEN\_ 0x07df

#define SNOW\_ 0x7fbc

#define ROSY\_BROWN\_ 0x5a31

#define TROPICAL\_ORANGE\_ 0x7e80

#define COCOUNT\_BROWN\_ 0x4102

#define SAND\_BROWN\_ 0x766b

#define RED\_ 0x7c00

#define BLUE\_ 0x001f

#define WHITE\_ 0x7fff

#define VIOLET\_ 0x6471

#define GAINSBORO\_ 0x6b5a

#define DARK\_GRAY\_ 0x0f7b

#define SALMON\_ 0x79ed

#define BLACK\_ 0x8000

#define LIGHT\_YELLOW 0x7FFE

#define LIGHT\_SALMON 0x7E52

#define LIGHT\_CORAL 0x7D28

#define YELLOW\_ 0x7FF0

#define WHEAT\_ 0x7775

#define BROWN\_ 0x50a5

#define PEACH\_PUFF\_ 0x7f18

#define TANGERINE\_ 0x7ea0

#define NAVAJO\_WHITE\_ 0x1F75

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_detect\_h\_

#define \_detect\_h\_

#define CHINESE 0

#define CALCULATE 1

#define ENGLISH 2

int detect(char\* input,int\* strategy);

int ddetect(char\* input);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_dog\_h\_

#define \_dog\_h\_

void dog\_body(int x,int y,float k,int color,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_printbase\_h\_

#define \_printfbase\_h\_

#define XMAX 1024

#define YMAX 768

/\*

typedef struct

{

unsigned char TRANSPARENT;

unsigned char BLUE;

unsigned char GREEN;

unsigned char RED;

}COLORBIN;

\*/

void start\_SVGA();

int examine\_start\_SVGA();

void pageturn(register int page);

int point\_drawing(int x,int y,int color,int page);

int picture\_input(int x,int y,const char\* path);

int line\_drawing(int x1,int y1,int x2,int y2,int color,int page,int status);//status=1,则为对话框绘制

void circle\_without\_drawing(int x,int y,int r,int color,int page);

void circle\_with\_drawing(int x,int y,int r,int color,int page,int status);

void exit\_SVGA();

void fill\_color(int x1,int y1,int x2,int y2,int color,int page,int status);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_English\_h\_

#define \_English\_h\_

#define CHAR\_WIDTH 24

#define CHAR\_HEIGHT 32

void English\_drawing(int x,int y,char c,int color,int page,int status);

void English\_text\_drawing(int x,int y,const char \*text,int color,int page,int status);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_fenci\_h\_

#define \_fenci\_h\_

#define MI\_VALUE 0.0000000291

#define ROW\_LENGTH 87921//3? 1

#define EACH\_LENGTH 13

#define INPUT\_SIZE 101

#define GB2312\_NUMBER 6763

int fenci(char \*input,char \*eventual);

double mi\_value(int i,int j);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_find\_h\_

#define \_find\_h\_

#include<lasthao.h>

char\* find\_logic(ZHUWEIBIN arr,char\* eventual,char\* result);

char\* file\_zhu(ZHUWEIBIN arr);

char\* file\_wei(ZHUWEIBIN arr);

char\* file\_bin(ZHUWEIBIN arr);

char\* weiyuci(ZHUWEIBIN arr);

char\* binyuci(ZHUWEIBIN arr);

int panduanNULL(ZHUWEIBIN arr);

char\* shenmeci(ZHUWEIBIN arr);

int panduanshenmeci(ZHUWEIBIN arr);

void replace(char \*str,const char \*s1,const char \*s2);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_fish\_h\_

#define \_fish\_h\_

void fish(int x,int y,float k,int color,int page);

void fish\_head(int x,int y,float k,int color,int page);

void fish\_eye(int x,int y,float k,int color,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_huatu1\_h\_

#define \_huatu1\_h\_

void background(int x,int y,int page);

void yellow\_head(int x,int y,int page);

void brown\_head(int x,int y,int page);

void black\_head(int x,int y,int page);

void white\_head(int x,int y,int page);

void black\_eye(int x,int y,int page);

void blue\_eye(int x,int y,int page);

void red\_eye(int x,int y,int page);

void nose(int x,int y,int page);

void smile\_mouth(int x,int y,int page);

void laugh\_mouth(int x,int y,int page);

void sad\_mouth(int x,int y,int page);

void boy\_hair(int x,int y,int page);

void girl\_hair(int x,int y,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_hz\_h\_

#define \_hz\_h\_

//以下头文件如总头文件中已包含则不用，在hz.c中包括头文件即可

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<graphics.h>

#include<math.h>

#include<bios.h>

#include<conio.h>

#include<dos.h>

#include<mouse.h>

#include<input.h>

#include<draw.h>

#include<color.h>

#include<xiezi.h>

#include<English.h>

#define NUM0 0x5230

#define NUM1 0x4f31

#define NUM2 0x5032

#define NUM3 0x5133

#define NUM4 0x4134

#define NUM5 0x4c35

#define NUM6 0x4d36

#define NUM7 0x4737

#define NUM8 0x4838

#define NUM9 0x4939

#define F2 0x3c00

#define F3 0x3d00

#define F4 0x3e00

#define F5 0x3f00

#define F6 0x4000

#define F7 0x4100

#define F8 0x4200

#define F9 0x4300

#define F10 0x4400

#define F1 0x3b00

#define ENTER 0x1c0d

#define BACK 0x0e08

#define ESC 0x011b

#define UP 0x4800

#define DOWN 0x5000

#define RIGHT 0x4d00

#define LEFT 0x4b00

#define ONE 0x0231

#define TWO 0x0332

#define THREE 0x0433

#define FOUR 0x0534

#define FIVE 0x0635

#define SIX 0x0736

#define SEVEN 0x0837

#define EIGHT 0x0938

#define NINE 0x0a39

#define ZERO 0x0b30

#define DELETE 0x5300

#define SPACE 0x3920

#define LASTLINE 0x1a5b

#define NEXTLINE 0x1b5d

#define FIRST 0x0231

#define SECOND 0x0332

#define THIRD 0x0433

#define FOURTH 0x0534

#define MAXPY 8

#define FAIL 0

#define SUCCESS 1

#define KEY\_A 0x1e61

#define KEY\_D 0x2064

int input\_method(int x, int y, char \*str, int value, char \*py, int page, int\* choose, int \*keyboard, int\* pianyi,char\* inputinput,int counting,char input[51][421],int\* input\_counter);

char \*itostr(int a,char \*s);

void pyFrm(int x1,int y1,int x2,int y2,int page);

void memory\_hz(int x,int y,int color[201][41],int page);

void clean\_hz(int x,int y,int color[201][41],int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_include\_h\_

#define \_include\_h\_

#include<graphics.h>

#include<page.h>

#include<color.h>

#include<draw.h>

#include<mouse.h>

#include<stdio.h>

#include<stdlib.h>

#include<page.h>

#include<dos.h>

#include<pstat.h>

#include<alloc.h>

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_input\_h\_

#define \_input\_h\_

int visible\_input(char \*str,int x,int y,int limit,int color,int color2,int page);

int invisible\_input(char \*str,int x,int y,int limit,int color,int color2,int page);

int visible\_input1(char \*str,int x,int y,int limit,int color,int color2,int page);

int invisible\_input1(char \*str,int x,int y,int limit,int color,int color2,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_calculate\_h\_

#define \_calculate\_h\_

void reverse(char \*str);

int compare(char \*num1,char \*num2);

void add(char \*num1,char \*num2,char \*result);

void subtract(char \*num1,char \*num2,char \*result);

void real\_add(char \*num1,char \*num2,char \*result);

void multiply(char \*num1,char\*num2,char \*result);

void real\_multiply(char \*num1, char \*num2, char \*result);

void reciprocal(char \*input,char \*output);

int isPrime(long long int num);

long long int gcd(long long int a,long long int b);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_lasthao\_h\_

#define \_lasthao\_h\_

#include"stce.h"

#define ZHUYU 0

#define BINYU 1

struct ZHUWEIBIN

{

int zhu[30];

int wei[30];

int bin[30];

};

void lasthao(char far\* input,char far\* fencihou,ZHUWEIBIN\* record);

void yijuhua(char\* temp,Tense,Tense,int beibei,ZHUWEIBIN\* zhuweibin);

int yiwen\_gantan(char\* str);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_logic\_h\_

#define \_logic\_h\_

#include<lasthao.h>

int virtual\_word(ZHUWEIBIN\* arr,ZHUWEIBIN\* virtual\_arr);

void virtual\_logic(ZHUWEIBIN\* arr,ZHUWEIBIN\* virtual\_arr);

int zhuweibin\_same(int\* arr1,int\* arr2);

int pron(ZHUWEIBIN arr);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_mouse\_h\_

#define \_mouse\_h\_

void mouseinit(int page);

void mouse\_drawing(int x,int y,int page);

void mread(int \*nx,int \*ny,int \*nbuttons);

void newmouse(int \*nx,int \*ny,int \*nbuttons,int page);

void save\_bk\_mou(int nx,int ny);

void clrmous(int nx,int ny,int page);

void drawmous(int nx,int ny,int page);

void UpdateMouse(int forceRefresh,int page);

int mouse\_press(int x1,int y1,int x2,int y2);

void memory\_mouse(int x,int y,int page);

void truly\_clean\_mouse(int x,int y,int page);

int get\_pixel\_color(int x, int y, int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_new\_h\_

#define \_new\_h\_

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PHR结构体：用于存储断句后句子中各个成分的标号，以及替代转化之后的标号

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef struct

{

char\* str; //存储段句后的这个短语

int num\_arr[15]; //存储用小数点分隔后的标号，如果没有找到num\_arr[0]==0

int num\_pre[15]; //存储一旦可以被关键词替代，替代后的词语的标号

int x; //用于判断该词是否需要关键词替代，x=1则可以被替代，x=0则不能被替代

int record;//表示num\_pre里面存了多少个数字

} PHR;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Temp结构体：用于临时存储搜索文件时暂时搜到的标号+短语或词语

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef struct

{

char num[20];

char phrase[100];

} Temp;

int atoi\_f(char\* a, int\* num\_arr, int n, int ci\_xing, int\* c);

int atoI(const char\* str);

int get\_num(char\* a);

PHR\* remark(char\*\* str);

PHR\* mark(char\*\* str, int\* ceng);

PHR\* input(char\*\* word,char\* biaohao);

PHR\* input2(char word[1][101],char\* biaohao);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_page\_h\_

#define \_page\_h\_

enum PAGE

{

PAGE\_TEST,

PAGE\_START,

PAGE\_REGISTER,

PAGE\_CHAT1,

PAGE\_CHAT2,

PAGE\_BEFORE,//返回上个界面

PAGE\_EXIT,//结束界面

PAGE\_CLOSE// 关闭界面

};

void change\_page(int \*page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_pchat\_h\_

#define \_pchat\_h\_

#define EXECUTE 0

#define FORBIDDEN 1

int page\_chat1();

int page\_SVGA\_chat1(int page,int keyboard,char input[51][421]);

void color\_change\_chat1(int i,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_pic\_h

#define \_pic\_h

//创建每个基本元素的结构体

typedef struct {

//下面是这个图形元素初始的性质

char name[30];//记录名称

//num[0]为该基本元素的第一层分类

//num[1]为该基本元素的第二层分类

//搜索的时候重点运用这个来搜索

int num[3];

//初始固定点的横纵坐标（就是第一笔画的那个起始点）

int x;

int y;

//下面是画图的时候根据要求可调的

//指定后的横纵坐标

//也就是指定的确切位置

int x1;

int y1;

//int color;//图形颜色

//判断大小，就是放缩的倍数k

int size;

//存储该元素的优先级

//数值越大优先级越高，越靠后显示

int level;

} object;

//创建节点，用于搜索生成什么图片

//用于查找时较为方便

typedef struct

{

object a;//二级结构体

int kind;//记录图像具体分类，降低查找的时间复杂度

} node;

void wen\_sheng\_tu(int color, int a[][5]);

void creat\_pic(object p,char name[30],int num[3],int x,int y);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_pregi\_h\_

#define \_pregi\_h\_

int register\_picture(int page);

void register\_color\_change\_mouse(int i,int page);

int check\_law(char \*input,int page);

int check\_password(char \*arr1,char \*arr2,int page);

int check\_password1(char \*arr1,char \*arr2,int page);

void text\_register(char \*arr1,char \*arr2);

void color\_change\_back\_mouse(int i,int page);

void back\_register\_color(int i,int page);

int page\_test();

int page\_chat2();

int page\_exit();

void clean\_chat1();

void clean\_chat2();

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_pstat\_h\_

#define \_pstat\_h\_

#define BIG\_NUMBER 10000

#define LEGAL\_INPUT 0

#define ILLEGAL\_INPUT 1

#define TEST\_INPUT 2

typedef struct

{

int num;//用户代号

char name[15];//用户名

char password[15];//密码

int chattext;//对话对应的文件编号

}LOGIN;

int page\_start(int page);

void mouse\_page\_start(int page,int\* num);

void color\_change\_mouse(int i,int page);

void string\_inspection(int page);

int password\_judge(char \*str,char \*password\_record,int \*n,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_ptext\_h\_

#define \_ptext\_h\_

#define USER 0

#define ROBOT 1

#define WORDWIDTH 32

#define TEXTWIDTH 704

#define TEXTWORD 22

#define ELLIPSEA 50

#define ELLIPSEB 55

#define FILL\_STATUS 0

#define NONFILL\_STATUS 1//bug?可能需要第三个标志位，不然会不会如果没有呈现，就大水漫灌了?

#define NONNONFILL\_STATUS 2

void robot\_avatar(int x,int y,int page);

void user\_avatar(int x,int y,int page);

void text\_rectangle(int ROM,int mode,int page,int y,int sum,int status);

void draw\_final0(char \*diaplay,int height,int y,int page,int status);

void draw\_final1(char \*diaplay,int height,int y,int page,int status);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_queue\_h\_

#define \_queue\_h\_

struct QueueNode

{

struct TrieNode\* data;

struct QueueNode\* next;

};

//链表形式，知道自己是谁，知道下一个是谁

typedef struct

{

QueueNode\* front;//队列之首

QueueNode\* rear;//队列之尾

}Queue;

//链表形式，知道第一个是谁，也知道最后一个是谁

void initqueue(Queue\* q);

void push(Queue\* q,struct TrieNode\* node);

struct TrieNode\* pop(Queue\* q);

int queue\_empty(Queue\* q);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_stceAC\_h\_

#define \_stceAC\_h\_

#define PAST 0

#define NOW 1

#define FUTURE 2

typedef enum

{

TENSE\_PAST,

//完成

TENSE\_PRESENT,//我们默认为这个时态

//在做

TENSE\_FUTURE,

//要去做

TENSE\_NONE

}Tense;

struct TrieNode

{

struct TrieNode\* children[53];//子节点

//话说一个TrieNode是有128个儿子，NULL的不行，不空才是好儿子(bushi

struct TrieNode\* failure;//失败指针

//这个指向失败后该去往的地方

Tense tense;//当前时态

//时态问题，这个需要次次记录次次调整

int end;//关键词结束标志

//结束标志，这个是末尾才变的

};

int tense(char\*temp);

int already(char\* temp);

int beidong(char\* str);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_teshu\_h\_

#define \_teshu\_h\_

#define FAILTESHU 0

#define SUCCESSTESHU 1

int teshu\_find(char \*input,char\* output);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_trans12\_h\_

#define \_trans12\_h\_

void trans12(char \*eventual,char \*biaohao);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_view\_h\_

#define \_view\_h\_

void mountain(int x,int y,float k,int color,int page);

void cloud(int x,int y,float k,int color,int page);

void water(int x,int y,float k,int color,int page);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_wentu\_h

#define \_wentu\_h\_

void wenshengtu(char\* shuru,char input[51][421]);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_xiezi\_h\_

#define \_xiezi\_h\_

int SVGA\_string\_length(const char \*str,int Chinese\_length,int English\_length);

void string\_drawing(const char \*str,int x,int y,int Chinese\_length,int English\_length,int size,int deviation,int color,int width,int length,int page,int status);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. **source函数<.c>**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<graphics.h>

#include<draw.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<color.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<stdlib.h>

#include<string.h>

//创建图形元素,并完善图形元素中的信息

//后面函数调用它时，只需先建立结构体，之后再调用这个函数，就可以完善结构体了。

void creat\_pic(object p,char name[30],int num[3],int x,int y)

{

int i=0;

//逐一赋值

strcpy(p.name,name);

for(i=0;i<3;i++)

{

p.num[i]=num[i];

}

p.x=x;

p.y=y;

//赋值完毕，可以按照所画的图来逐一完善结构体

}

//在已知图形中元素及位置时将具体位置根实际情况据有机调整

//将各个部分的图像有机合并

//重点是优化升级，进而避免图形的错位

void merge(object\* a)

{

int i=0;

int j=0;

switch(a[i])

{

case 0:

}

}

//画出最终生成的图

void wen\_sheng\_tu(int color[4],int a[5],int b[5],int c[5])

{

int a[8];

int draw;

int i;

int j;

int p;

//按要求创建背景

//背景最多2个

fill\_color(0,0,1024,768,color,page,0);//最后一位是0/1不一定

switch()

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

/\*文件名称：pregi

/\*文件目的：实现注册界面

/\*时间复杂度：\Theta(1)

/\*空间复杂度：\Theta(1024\*768)

\*/

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<draw.h>

#include<color.h>

#include<page.h>

#include<mouse.h>

#include<dos.h>

#include<time.h>

#include<bios.h>

#include<xiezi.h>

#include<pstat.h>

#include<English.h>

#include<ptext.h>

#include<pregi.h>

#include<input.h>

#include<pchat.h>

extern int MouseX,MouseY,MouseS,press;

extern void\* buffer;

extern union REGS our\_AI\_regs;

extern int MouseX\_memory,MouseY\_memory;

//函数名称：register\_picture

//函数目的：实现注册界面

int register\_picture(int page)

{

//mread(&MouseX,&MouseY,&MouseS);

//clrmous(MouseX,MouseY,page);

delay(20);

//save\_bk\_mou(MouseX,MouseY);

//鼠标部分

fill\_color(0,0,1024,768,SPRING\_GREEN\_,page,0);

robot\_avatar(55,35+80,page);

int i;

for(i=-1;i<=1;i++)

{

line\_drawing(104,35+80,135,15+80,WHITE\_,page,1);

line\_drawing(104,35+80,135,40+80,WHITE\_,page,1);

line\_drawing(135,15+80,475,15+80,WHITE\_,page,1);

}

fill\_color(135,15+80,475,15+80+WORDWIDTH,WHITE\_,page,1);

for(i=16;i<40;i++)

{

line\_drawing(104,35+80,135,i+80,WHITE\_,page,1);

}

//else if(status==NONFILL\_STATUS)

//{

// fill(120,1,WHITE\_,page,1);

// ;

//}

line\_drawing(104,35+80,135,15+80,BLACK\_,page,1);

line\_drawing(104,35+80,135,40+80,BLACK\_,page,1);

line\_drawing(135,15+80,400,15+80,BLACK\_,page,1);

line\_drawing(135,15+80,475,15+80,BLACK\_,page,1);

line\_drawing(475,15+80,475,15+80+WORDWIDTH,BLACK\_,page,1);

line\_drawing(475,15+80+WORDWIDTH,135,15+80+WORDWIDTH,BLACK\_,page,1);

line\_drawing(135,20+80,135,15+80,BLACK\_,page,1);

line\_drawing(135,40+80,135,15+80+WORDWIDTH,BLACK\_,page,1);

string\_drawing("请注册！",140,13+80,32,20,12,-8,RED\_,32\*23,10000,page,1);

string\_drawing("用户名：",196,290,32,20,12,-8,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

string\_drawing("密码：",228,410,32,20,12,-8,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

string\_drawing("再次确认密码：",100,530,32,20,12,-8,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

fill\_color(350,260,900,360,TROPICAL\_ORANGE\_,page,0);

fill\_color(350,380,900,480,TROPICAL\_ORANGE\_,page,0);

fill\_color(350,500,900,600,TROPICAL\_ORANGE\_,page,0);

fill\_color(300,650,600,720,TROPICAL\_ORANGE\_,page,0);

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

string\_drawing("注册",400,670,32,20,12,-8,WHITE\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

//以上是画图部分

int status=0;

int position=0;

char name\_input[16]={'\0'};

char password\_input[16]={'\0'};

char password\_again\_input[16]={'\0'};

int error1;

int error2;

int error3;

int error4;

mouseinit(page);

while(1)

{

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX,MouseY,page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX,MouseY,page);

delay(20);

truly\_clean\_mouse(MouseX,MouseY,page);

//退出

if(MouseX>=964&&MouseX<=1024&&MouseY>=0&&MouseY<=60)

{

if(mouse\_press(964,0,1024,60)==2)

{

//if(status==0)

{

MouseS=1;

status=4;

}

continue;

}

else if(mouse\_press(964,0,1024,60)==1)

{

register\_color\_change\_mouse(4,page);

status=4;

MouseS=0;

page=PAGE\_START;

change\_page(&page);

}

}

//用户名

else if(MouseX>=350&&MouseX<=900&&MouseY>=260&&MouseY<=360)

{

if(mouse\_press(350,260,900,360)==2)

{

MouseS=2;

if(status==0&&position==0)

{

status=1;

}

continue;

}

else if(mouse\_press(350,260,900,360)==1)

{

status=1;

MouseS=0;

register\_color\_change\_mouse(1,page);

name\_input[0]='\0';

visible\_input(name\_input,350,260,15,BLACK\_,DARK\_GRAY\_,page);

error1=check\_law(name\_input,page);

continue;

}

}

//密码

else if(MouseX>=350&&MouseX<=900&&MouseY>=380&&MouseY<=480)

{

if(mouse\_press(350,380,900,480)==2)

{

MouseS=2;

if(status==0&&position==0)

{

status=2;

}

continue;

}

else if(mouse\_press(350,380,900,480)==1)

{

status=2;

MouseS=0;

register\_color\_change\_mouse(2,page);

password\_input[0]='\0';

invisible\_input(password\_input,350,380,15,BLACK\_,DARK\_GRAY\_,page);

error2=check\_law(password\_input,page);

error4=check\_password1(password\_again\_input,password\_input,page);

continue;

}

}

//再次输入密码

else if(MouseX>=350&&MouseX<=900&&MouseY>=500&&MouseY<=600)

{

if(mouse\_press(350,500,900,600)==2)

{

MouseS=2;

if(status==0&&position==0)

{

status=3;

}

continue;

}

else if(mouse\_press(350,500,900,600)==1)

{

status=3;

MouseS=0;

register\_color\_change\_mouse(3,page);

password\_again\_input[0]='\0';

invisible\_input(password\_again\_input,350,500,15,BLACK\_,DARK\_GRAY\_,page);

error3=check\_law(password\_again\_input,page);

error4=check\_password(password\_again\_input,password\_input,page);

continue;

}

}

//注册区域

else if(MouseX>=300&&MouseX<=650&&MouseY>=600&&MouseY<=720)

{

if(mouse\_press(300,650,600,720)==2)

{

MouseS=1;

if(status==0&&position==0)

{

status=1;

}

continue;

}

else if(mouse\_press(300,650,600,720)==1)

{

MouseS=0;

register\_color\_change\_mouse(5,page);

if(error1==0&&error2==0&&error3==0&&error4==0)

{

text\_register(name\_input,password\_input);

page=PAGE\_START;

fill\_color(600,55,600+48\*5,55+48,SPRING\_GREEN\_,page,0);

string\_drawing("注册成功！",600,55,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

change\_page(&page);

}

else

{

string\_drawing("注册有误！",600,55,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

fill\_color(350,260,900,360,TROPICAL\_ORANGE\_,page,0);

fill\_color(350,380,900,480,TROPICAL\_ORANGE\_,page,0);

fill\_color(350,500,900,600,TROPICAL\_ORANGE\_,page,0);

fill\_color(300,650,600,720,TROPICAL\_ORANGE\_,page,0);

string\_drawing("注册",400,670,32,20,12,-8,WHITE\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

}

continue;

}

}

//其他区域

else

{

/\*

switch(status)

{

case 1:

back\_register\_color(1,page);

break;

case 2:

back\_register\_color(2,page);

break;

case 3:

back\_register\_color(3,page);

break;

}

\*/

MouseS=0;

status=4;

}

}

return PAGE\_REGISTER;

}

//函数名称：register\_color\_change\_mouse

//函数目的：实现注册页的鼠标改图标颜色

void register\_color\_change\_mouse(int i,int page)

{

//clrmous(MouseX,MouseY,page);

delay(10);

switch(i)

{

//用户名

case 1:

fill\_color(350,260,900,360,DARK\_GRAY\_,page,0);

break;

//密码

case 2:

fill\_color(350,380,900,480,DARK\_GRAY\_,page,0);

break;

//再次确认密码

case 3:

fill\_color(350,500,900,600,DARK\_GRAY\_,page,0);

break;

//退出

case 4:

fill\_color(964,0,1024,60,DARK\_GRAY\_,page,0);

break;

//注册

case 5:

fill\_color(300,650,600,720,DARK\_GRAY\_,page,0);

break;

}

}

//函数名称：check\_law

//函数目的：判断用户名和密码是否符合命名规范

int check\_law(char \*input,int page)

{

int length=strlen(input);

int i;

if(length>=15)

{

string\_drawing("最多十五位！",550,175,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

return 1;

}

for(i=0;i<length;i++)

{

if((input[i]>='0'&&input[i]<='9')||(input[i]>='a'&&input[i]<='z')||(input[i]>='A'&&input[i]<='Z'))

{

//无事发生

}

else

{

string\_drawing("只能输入字母和数字！",100,75,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

input[i]='\0';

return 1;

}

}

return 0;

//0是没问题，1是有问题

}

//函数名称：check\_password

//函数目的：检查密码是否再次输入与首次输入一致

int check\_password(char \*arr1,char \*arr2,int page)

{

if(strcmp(arr1,arr2)==0)

{

return 0;

}

else

{

string\_drawing("两次密码不一致！",550,115,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

return 1;

}

//同样地，0是没问题，1是有问题

}

//函数名称：check\_password1

//函数目的：检查密码是否再次输入与首次输入一致

int check\_password1(char \*arr1,char \*arr2,int page)

{

if(strcmp(arr1,arr2)==0)

{

return 0;

}

else

{

//string\_drawing("两次密码不一致！",550,115,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

return 1;

}

//同样地，0是没问题，1是有问题

}

//函数名称：text\_register

//函数目的：把信息写入文件

void text\_register(char \*arr1,char \*arr2)

{

LOGIN\* log;

FILE\* fp;

if((fp=fopen("C:\\CODE\\data\\password.dat","rb+"))==NULL)

{

printf("密码数据库无法打开！任意键退出");

getchar();

exit(1);

}

if((log=(LOGIN\*)malloc(sizeof(LOGIN)))==NULL)//分配空间

{

printf("密码内存分配不足！任意键退出");

getchar();

exit(1);

}

fseek(fp,0,SEEK\_END);

int sum=ftell(fp)/sizeof(LOGIN);

log->num=sum+1;

strcpy(log->name,arr1);

strcpy(log->password,arr2);

log->chattext=0;

fseek(fp,0,SEEK\_END);

fwrite(log,sizeof(LOGIN),1,fp);

if(log!=NULL)

{

free(log);

log=NULL;

}

if(fclose(fp)!=0)

{

printf("密码数据库文件关不掉！任意键退出");

getchar();

exit(1);

}

}

//函数名称：back\_register\_color

//函数目的：将变化颜色的注册按键复原

void back\_register\_color(int i,int page)

{

switch(i)

{

case 1:

fill\_color(350,260,900,360,TROPICAL\_ORANGE\_,page,0);

break;

case 2:

fill\_color(350,380,900,480,TROPICAL\_ORANGE\_,page,0);

break;

case 3:

fill\_color(350,500,900,600,TROPICAL\_ORANGE\_,page,0);

break;

}

}

int page\_test()

{

return PAGE\_TEST;

}

int page\_chat2()

{

return PAGE\_CHAT2;

}

int page\_exit()

{

return PAGE\_EXIT;

}

void clean\_chat1()

{

}

void clean\_chat2()

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名称：pstat

\*\* 文件目的：实现登录页面的画图

\*\* 时间复杂度：Bresenham算法对应的复杂度

\*\* 空间复杂度：储存的页面像素大小

\*/

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<draw.h>

#include<color.h>

#include<page.h>

#include<mouse.h>

#include<dos.h>

#include<time.h>

#include<bios.h>

#include<xiezi.h>

#include<pstat.h>

#include<English.h>

#include<pregi.h>

#include<input.h>

#include<Bezier.h>

#include<ptext.h>

extern int MouseX,MouseY,MouseS,press;

extern void\* buffer;

extern union REGS our\_AI\_regs;

extern int MouseX\_memory,MouseY\_memory;

//函数名称：page\_start

//函数目的：实现登录

int page\_start(int page)

{

//左上角的小机器人

int i;

fill\_color(0,0,1024,768,SPRING\_GREEN\_,page,0);

circle\_with\_drawing(224,210,140,SNOW\_,page,0);

fill\_color(154,138,294,212,ROSY\_BROWN\_,page,0);

circle\_with\_drawing(192,176,20,YELLOW\_,page,0);

circle\_with\_drawing(256,176,20,YELLOW\_,page,0);

for(i=-3;i<=3;i++)

{

line\_drawing(192+i,138,192+i,114,COCOUNT\_BROWN\_,page,0);

line\_drawing(256+i,138,256+i,114,COCOUNT\_BROWN\_,page,0);

}//稍微粗一点

circle\_with\_drawing(192,101,13,SPRING\_GREEN\_,page,0);

circle\_with\_drawing(256,101,13,SPRING\_GREEN\_,page,0);

fill\_color(192,212,256,338,ROSY\_BROWN\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(192,256+i,154,256+i,ROSY\_BROWN\_,page,0);

line\_drawing(256,256+i,294,256+i,ROSY\_BROWN\_,page,0);

}

for(i=0;i<5;i++)

{

line\_drawing(154-i,226,154-i,286,ROSY\_BROWN\_,page,0);

line\_drawing(294+i,226,294+i,286,ROSY\_BROWN\_,page,0);

line\_drawing(154,226-i,129,226-i,ROSY\_BROWN\_,page,0);

line\_drawing(154,286+i,129,286+i,ROSY\_BROWN\_,page,0);

line\_drawing(294,226-i,319,226-i,ROSY\_BROWN\_,page,0);

line\_drawing(294,286+i,319,286+i,ROSY\_BROWN\_,page,0);

}

circle\_with\_drawing(224,252,29,LIGHT\_SALMON,page,0);

fill\_color(192,212,256,252,ROSY\_BROWN\_,page,0);

circle\_with\_drawing(256,176,5,RED\_,page,0);

circle\_with\_drawing(192,176,5,RED\_,page,0);

fill\_color(206,252,218,264,WHITE\_,page,0);

fill\_color(230,252,242,264,WHITE\_,page,0);

//右上角的文字

string\_drawing("您的智能助手：小智",440,150,48,26,15,0,RED\_,4000,12,page,0);

//string\_drawing("AI",850,150,32,20,12,-8,RED\_,400,12,page,0);

//string\_drawing("produced by",530,250,24,15,9,-6,RED\_,348,9,page,0);

//string\_drawing("作者：孟宪哲，徐多泽",480,250,24,15,9,-6,RED\_,1000,12,page,0);

string\_drawing("作者：孟",480,250,24,15,9,-6,RED\_,1000,12,page,0);

string\_drawing("洁",480+5\*24+4,250,24,15,9,-6,RED\_,1000,12,page,0);

fill\_color(480+5\*24+4,250,480+5\*24+12,300,SPRING\_GREEN\_,page,0);

string\_drawing("洁",480+5\*24-9,250,24,15,9,-6,RED\_,1000,12,page,0);

fill\_color(480+5\*24-8,250,480+5\*24-1,300,SPRING\_GREEN\_,page,0);

string\_drawing("宪",480+4\*24,250,24,15,9,-6,RED\_,1000,12,page,0);

//fill\_color(480+5\*24,250,480+5\*24+8,300,SPRING\_GREEN\_,page,0);

//string\_drawing("颉",480+5\*24,253,24,15,9,-6,RED\_,1000,12,page,0);

//fill\_color(480+5\*24+12,250,480+5\*24+24,300,SPRING\_GREEN\_,page,0);

//fill\_color(480+5\*24+20,250,480+5\*24+32,300,SPRING\_GREEN\_,page,0);

string\_drawing("，徐多泽",480+6\*24+4,250,24,15,9,-6,RED\_,1000,12,page,0);

//下面的框

string\_drawing("用户名：",100,440,32,20,12,-8,BLUE\_,128,12,page,0);

string\_drawing("密码：",120,580,32,20,12,-8,BLUE\_,96,12,page,0);

fill\_color(310,420,850,510,TROPICAL\_ORANGE\_,page,0);//用户名

fill\_color(310,560,850,650,TROPICAL\_ORANGE\_,page,0);//密码

fill\_color(150,680,480,750,TROPICAL\_ORANGE\_,page,0);//登录

fill\_color(530,680,860,750,TROPICAL\_ORANGE\_,page,0);//注册

string\_drawing("登录",260,700,32,20,12,-8,BLUE\_,64,12,page,0);

string\_drawing("注册",645,700,32,20,12,-8,BLUE\_,64,12,page,0);

//该部分画完之后如果再改，也要改下面的登录注册代码数据

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

//画右上角的叉号

while(1)

{

mouse\_page\_start(page,&i);

delay(20);

}

return PAGE\_START;

}

//函数名称：mouse\_page\_start

//函数目的：实现登录注册界面鼠标的移动和点击跳转

void mouse\_page\_start(int page,int \*num)

{

//mread(&MouseX,&MouseY,&MouseS);

//memory\_mouse(MouseX,MouseY);

mouseinit(page);

//truly\_clean\_mouse(MouseX,MouseY);

//clrmous(MouseX,MouseY,page);

//delay(20);

//save\_bk\_mou(MouseX,MouseY);

int status=0;

int position=0;//判断是否里面有字

int temp=TEST\_INPUT;

char str[16]={0};

char password\_recording[16]={0};

char str\_input[16]={0};

char password\_recording\_input[16]={0};

while(1)

{

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX,MouseY,page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX,MouseY,page);

delay(20);

truly\_clean\_mouse(MouseX,MouseY,page);

//用户名

if(MouseX>=310&&MouseX<=850&&MouseY>=420&&MouseY<=510)

{

if(mouse\_press(310,420,850,510)==2)

{

MouseS=2;

if(status==0&&position==0)

{

status=1;//bug:status!=1

}

continue;

}

else if(mouse\_press(310,420,850,510)==1)

{

status=1;

color\_change\_mouse(1,page);

MouseS=0;

//memset(str\_input,0,16);

visible\_input1(str\_input,310,420,15,BLACK\_,GAINSBORO\_,page);

//str\_input[0]='\0';

if(temp==ILLEGAL\_INPUT)

{

string\_inspection(page);

temp=TEST\_INPUT;

page=0;

//break;

}

if(temp==LEGAL\_INPUT)

{

if(strlen(str\_input)!=0)

{

position=1;

}

else

{

position=0;

}

temp=TEST\_INPUT;

continue;

}

}

}

//密码

else if(MouseX>=310&&MouseX<=850&&MouseY>=560&&MouseY<=650)

{

if(mouse\_press(310,560,850,650)==2)

{

MouseS=2;

if(status==0&&position==0)

{

status=2;

}

continue;

}

else if(mouse\_press(310,560,850,650)==1)

{

status=2;

color\_change\_mouse(2,page);

MouseS=0;

invisible\_input1(password\_recording\_input,310,560,15,BLACK\_,GAINSBORO\_,page);

str[0]='\0';

if(temp==ILLEGAL\_INPUT)

{

string\_inspection(page);

temp=TEST\_INPUT;

page=0;

//break;

}

if(temp==LEGAL\_INPUT)

{

if(strlen(str)!=0)

{

position=1;

}

else

{

position=0;

}

temp=TEST\_INPUT;

continue;

}

}

}

//登录

else if(MouseX>=150&&MouseX<=480&&MouseY>=680&&MouseY<=750)

{

if(mouse\_press(150,680,480,750)==2)

{

MouseS=1;

status=3;

if(status==0)

{

color\_change\_mouse(3,page);

status=3;

}

continue;

}

else if(mouse\_press(150,680,480,750)==1)

{

color\_change\_mouse(3,page);

//text\_register("flag1","flag2");//test

/\*

getchar();

string\_drawing(password\_recording\_input,0,0,48,26,15,0,TROPICAL\_ORANGE\_,BIG\_NUMBER,BIG\_NUMBER,page);

getchar();

exit(1);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*test\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

status=3;

MouseS=0;

if(password\_judge(str\_input,password\_recording\_input,num,page)==1)

{

fill\_color(0,0,1024,768,WHITE\_,page,0);

string\_drawing("登录成功！",274,220,48,26,15,0,TROPICAL\_ORANGE\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

delay(2000);

fill\_color(274,220,524,268,WHITE\_,page,0);

page=PAGE\_CHAT1;

change\_page(&page);//跳转至用户功能界面

break;

}

else

{

//fill\_color(270,225,370,255,DARK\_GRAY\_,page);

//string\_drawing("登录失败！",274,220,48,26,15,0,TROPICAL\_ORANGE\_,BIG\_NUMBER,BIG\_NUMBER,page);

//delay(20);

}

}

}

//注册

else if(MouseX>=530&&MouseX<=860&&MouseY>=680&&MouseY<=750)

{

if(mouse\_press(530,680,860,750)==2)

{

MouseS=1;

if(status==0)

{

color\_change\_mouse(4,page);

status=4;

}

continue;

}

else if(mouse\_press(530,680,860,750)==1)

{

MouseS=0;

color\_change\_mouse(4,page);

page=PAGE\_REGISTER;

change\_page(&page);

break;

}

}

//退出

else if(MouseX>=964&&MouseX<=1024&&MouseY>=0&&MouseY<=60)

{

if(mouse\_press(964,0,1024,60)==2)

{

MouseS=1;

if(status==0)

{

color\_change\_mouse(5,page);

status=5;

}

continue;

}

else if(mouse\_press(964,0,1024,60)==1)

{

color\_change\_mouse(5,page);

MouseS=0;

delay(200);

exit(1);

break;

}

}

//其他位置

else

{

switch(status)

{

/\*

case 1:

color\_change\_back\_mouse(1,page);

break;

case 2:

color\_change\_back\_mouse(2,page);

break;

\*/

case 3:

color\_change\_back\_mouse(3,page);

break;

}

MouseS=0;

//color\_change\_mouse(6,page);

status=6;

}

}

}

//函数名称：color\_change\_mouse

//函数目的：随鼠标的点击，界面发生变化

void color\_change\_mouse(int i,int page)

{

//clrmous(MouseX,MouseY,page);

delay(10);

switch(i)

{

case 1:

//用户名

fill\_color(310,420,850,510,GAINSBORO\_,page,0);

break;

case 2:

//密码

fill\_color(310,560,850,650,GAINSBORO\_,page,0);

break;

case 3:

//登录

fill\_color(150,680,480,750,GAINSBORO\_,page,0);

string\_drawing("登录",220,700,32,20,12,-8,BLUE\_,64,12,page,0);

break;

case 4:

//注册

fill\_color(530,680,860,750,GAINSBORO\_,page,0);

string\_drawing("注册",600,700,32,20,12,-8,BLUE\_,64,12,page,0);

break;

case 5:

//退出

fill\_color(964,0,1024,60,SALMON\_,page,0);

break;

default:

fill\_color(310,420,850,510,TROPICAL\_ORANGE\_,page,0);//用户名

fill\_color(310,560,850,650,TROPICAL\_ORANGE\_,page,0);//密码

fill\_color(150,680,480,750,TROPICAL\_ORANGE\_,page,0);//登录

fill\_color(530,680,860,750,TROPICAL\_ORANGE\_,page,0);//注册

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

break;

}

}

//函数名称：color\_change\_back\_mouse

//函数目的：将变化的框框变回来

void color\_change\_back\_mouse(int i,int page)

{

switch(i)

{

case 1:

fill\_color(310,420,850,510,TROPICAL\_ORANGE\_,page,0);

break;

case 2:

fill\_color(310,560,850,650,TROPICAL\_ORANGE\_,page,0);

break;

case 3:

fill\_color(150,680,480,750,TROPICAL\_ORANGE\_,page,0);

string\_drawing("登录",220,700,32,20,12,-8,BLUE\_,64,12,page,0);

break;

}

}

//函数名称：string\_inspection

//函数目的：字符串里面违法字符提示

void string\_inspection(int page)

{

string\_drawing("只能输入数字和字母",330,430,32,20,12,-8,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

}

//函数名称：password\_judge

//函数目的：判断密码是否正确

int password\_judge(char \*str,char \*password\_record,int \*n,int page)

{

int haoma;

int i;

FILE \*fp;

//LOGIN \*input\_now=NULL;

char input\_now[35];

char pass[17]={0};

char name[15]={0};

char read\_information[34];

char temp;

if((fp=fopen("C:\\CODE\\data\\password.dat","rb+"))==NULL)

{

printf("无法打开密码库！任意键退出");

getchar();

exit(1);

}

//获知文件多大

fseek(fp,0,SEEK\_END);

long int size=ftell(fp)/34;

int size\_record=0;

//进行字符串的比对

strcpy(input\_now," ");

for(i=2;i<18;i++)

{

input\_now[i]=str[i-2];

}

for(i=17;i<33;i++)

{

input\_now[i]=password\_record[i-17];

}

input\_now[33]=' ';

input\_now[34]='\0';

for(i=0;i<34;i++)

{

if((input\_now[i]>='a'&&input\_now[i]<='z')

||(input\_now[i]>='A'&&input\_now[i]<='Z')

||(input\_now[i]>='0'&&input\_now[i]<='9'))

{

;

}

else

{

input\_now[i]=' ';

}

}

//string\_drawing(input\_now,0,0,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page);//test

fseek(fp,0,SEEK\_SET);

while(1)

{

size\_record++;

for(i=0;i<34;i++)

{

temp=getc(fp);

read\_information[i]=temp;

if((read\_information[i]>='a'&&read\_information[i]<='z')

||(read\_information[i]>='A'&&read\_information[i]<='Z')

||(read\_information[i]>='0'&&read\_information[i]<='9'))

{

;

}

else

{

read\_information[i]=' ';

}

}

/\*

string\_drawing(read\_information,0,0,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page);

break;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*test\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

if(strcmp(input\_now,read\_information)==0)

{

fill\_color(530,300,730,350,SPRING\_GREEN\_,page,0);

string\_drawing("密码正确",530,300,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

for(i=0;i<strlen(input\_now);i++)

{

input\_now[i]=0;

}

for(i=0;i<strlen(read\_information);i++)

{

read\_information[i]=0;

}

return 1;

}

if(size\_record==size)

{

string\_drawing("密码错误",530,300,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

fill\_color(310,420,850,510,GAINSBORO\_,page,0);

fill\_color(310,560,850,650,GAINSBORO\_,page,0);

fill\_color(310,420,850,510,TROPICAL\_ORANGE\_,page,0);//用户名

fill\_color(310,560,850,650,TROPICAL\_ORANGE\_,page,0);//密码

fill\_color(150,680,480,750,TROPICAL\_ORANGE\_,page,0);//登录

fill\_color(530,680,860,750,TROPICAL\_ORANGE\_,page,0);//注册

string\_drawing("登录",220,700,32,20,12,-8,BLUE\_,64,12,page,0);

string\_drawing("注册",600,700,32,20,12,-8,BLUE\_,64,12,page,0);

for(i=0;i<16;i++)

{

str[i]=0;

}

for(i=0;i<16;i++)

{

password\_record[i]=0;

}

return 0;

}

}

/\*

fseek(fp,0,SEEK\_SET);//还是SEEK\_END?

haoma=ftell(fp)/34;//34是name+password

for(i=0;i<haoma;i++)

{

//if((input\_now=(LOGIN\*)malloc(sizeof(LOGIN)))==NULL)

//{

// printf("密码记录内存分配不足！任意键退出");

// getchar();

// exit(1);

//}

fseek(fp,i\*34,SEEK\_END);//34是name+password

fread(input\_now,34,1,fp);//34是name+password

for(i=2;i<=16;i++)

{

name[i-2]=input\_now[i];

}

for(i=17;i<=33;i++)

{

pass[i-17]=input\_now[i];

}

//if(strcmp(password\_record,input\_now->password)==0)

if(strcmp(password\_record,pass)==0)

{

//if(strcmp(str,input\_now->password)!=0)

if(strcmp(str,pass)!=0)

{

string\_drawing("密码错误",530,300,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page);

return 0;

}

else

{

//对了就返回1

//\*n=input\_now->num;

string\_drawing("密码正确",530,300,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page);

if(input\_now!=NULL)

{

free(input\_now);

//input\_now=NULL;

}

if(fclose(fp)!=0)

{

printf("没关掉密码记录文件！任意键退出");

getchar();

exit(1);

}

return 1;

}

}

}

if(i==haoma)

{

string\_drawing("账号未注册",530,300,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page);

return 0;

}

if(input\_now!=NULL)

{

free(input\_now);

//input\_now=NULL;

}

if(fclose(fp)!=0)

{

printf("没关掉密码记录文件！任意键退出");

getchar();

exit(1);

}

//不对就返回0

return 0;

//什么状态屏幕都有显示

\*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：ptext

\*\*文件目的：实现对话框的构建

\*\*时间复杂度：最小，但是正比于字符串长度

\*\*空间复杂度：\Theta(像素存储个数)

\*/

#include<Bezier.h>

#include<ptext.h>

#include<string.h>

#include<stdlib.h>

#include<alloc.h>

#include<stdio.h>

#include<draw.h>

#include<xiezi.h>

#include<color.h>

//extern char far input[21][421];//test

void robot\_avatar(int x,int y,int page)

//x,y是圆心位置

{

circle\_with\_drawing(x,y,50,WHITE\_,page,1);

fill\_color(x-37,y-22,x+41,y+30,GAINSBORO\_,page,1);

circle\_with\_drawing(x-24,y,15,YELLOW\_,page,1);

circle\_with\_drawing(x+24,y,15,YELLOW\_,page,1);

circle\_with\_drawing(x-24,y,4,RED\_,page,1);

circle\_with\_drawing(x+24,y,4,RED\_,page,1);

static int x0[3];

static int y0[3];

x0[0]=x-16;

y0[0]=y+22;

x0[1]=x;

y0[1]=y+32;

x0[2]=x+16;

y0[2]=y+22;

Bezier(RED\_,3,page,x0,y0,1);

circle\_with\_drawing(x-24,y-35,5,SPRING\_GREEN\_,page,1);

circle\_with\_drawing(x+24,y-35,5,SPRING\_GREEN\_,page,1);

line\_drawing(x-24,y-22,x-24,y-30,BROWN\_,page,1);

line\_drawing(x+24,y-22,x+24,y-30,BROWN\_,page,1);

}

void user\_avatar(int x,int y,int page)

//x,y是圆心位置

{

circle\_with\_drawing(x,y,50,YELLOW\_,page,1);

circle\_with\_drawing(x-30,y-20,15,WHITE\_,page,1);

circle\_with\_drawing(x+30,y-20,15,WHITE\_,page,1);

circle\_with\_drawing(x-30,y-20,4,BLACK\_,page,1);

circle\_with\_drawing(x+30,y-20,4,BLACK\_,page,1);

static int x0[3];

static int y0[3];

x0[0]=x-25;

y0[0]=y+10;

x0[1]=x;

y0[1]=y+50;

x0[2]=x+25;

y0[2]=y+10;

Bezier(RED\_,3,page,x0,y0,1);

}

void text\_rectangle(int RAM,int mode,int page,int y,int sum,int status)

//其中的y需要通过鼠标拖动的逻辑算出来

//字符串从文件里读取

//!!!此时的文件里面字母都是重复两遍的!!!

{

char paths[50]={0};

sprintf(paths,"C:\\CODE\\chat\\chat%d",RAM);

FILE \*fp;

fp=fopen(paths,"rt+");//可读可写

char display[10001];//支持5000汉字

char temp=1;

int count=0;

while(temp!=0)

{

temp=fgetc(fp);

display[count++]=temp;

}

count--;

//这时候，count就是字符串里面不算上\0的长度

//计算输入法框框的大小

int hanzishu=count/2-sum;

int ascshu=sum;

int height=(hanzishu+ascshu)/TEXTWORD+1;

if(mode==USER)

{

draw\_final0(display,height,y,page,status);

}

if(mode==ROBOT)

{

draw\_final1(display,height,y,page,status);

}

fclose(fp);

}

void draw\_final0(char \*display,int height,int y,int page,int status)

//y是起始点，可以为负数

{

//图像与对话框

robot\_avatar(55,y+35,page);

int i;

for(i=-1;i<=1;i++)

{

line\_drawing(104,y+35,135,y+15,WHITE\_,page,1);

line\_drawing(104,y+35,135,y+40,WHITE\_,page,1);

line\_drawing(135,y+15,800,y+15,WHITE\_,page,1);

}

fill\_color(135,y+15,950,y+15+height\*WORDWIDTH,WHITE\_,page,1);

for(i=16;i<40;i++)

{

line\_drawing(104,y+35,135,y+i,WHITE\_,page,1);

}

//else if(status==NONFILL\_STATUS)

//{

// fill(120,1,WHITE\_,page,1);

// ;

//}

line\_drawing(104,y+35,135,y+15,BLACK\_,page,1);

line\_drawing(104,y+35,135,y+40,BLACK\_,page,1);

line\_drawing(135,y+15,950,y+15,BLACK\_,page,1);

line\_drawing(950,y+15,950,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(950,y+15+height\*WORDWIDTH,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(135,y+20,135,y+15,BLACK\_,page,1);

line\_drawing(135,y+40,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

//显示汉字

int pointer=0;

int recordx=0;

int recordy=0;

char duanju[47]={0};

int jilu=0;

int yhuanhang=0;

int yzhendehuanhang=0;

int length=strlen(display);

char temp[3]={0};

int j,m=0;

for(i=0;i<length;i+=2)

{

temp[0]=display[i];

temp[1]=display[i+1];

if(temp[0]>0)

{

temp[1]='\0';

i--;

}

if(yhuanhang==23)

{

yhuanhang=0;

recordx=0;

yzhendehuanhang++;

}

else

{

yhuanhang++;

}

string\_drawing(temp,165+recordx,y+32\*yzhendehuanhang+15,32,20,12,-8,BLACK\_,23,23,page,1);

m++;

if(temp[0]>0)

{

recordx+=24;

}

else

{

recordx+=32;

}

if(recordx==TEXTWORD)

{

recordx=0;

recordy++;

}

}

}

void draw\_final1(char \*display,int height,int y,int page,int status)

//y是起始点，可以为负数

{

//图像与对话框

user\_avatar(969,y+35,page);

int i;

for(i=-1;i<=1;i++)

{

line\_drawing(920,y+35,889,y+20,WHITE\_,page,1);

line\_drawing(920,y+35,889,y+40,WHITE\_,page,1);

line\_drawing(889,y+15,224,y+15,WHITE\_,page,1);

}

fill\_color(142,y+15,889,y+15+height\*WORDWIDTH,WHITE\_,page,1);

for(i=21;i<40;i++)

{

line\_drawing(920,y+35,889,y+i,WHITE\_,page,1);

}

line\_drawing(920,y+35,889,y+20,BLACK\_,page,1);

line\_drawing(920,y+35,889,y+40,BLACK\_,page,1);

line\_drawing(889,y+15,224,y+15,BLACK\_,page,1);

line\_drawing(142,y+15,889,y+15,BLACK\_,page,1);

line\_drawing(889,y+15+height\*WORDWIDTH,142,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(142,y+15,142,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(889,y+15+height\*WORDWIDTH,889,y+40,BLACK\_,page,1);

line\_drawing(889,y+15,889,y+20,BLACK\_,page,1);

//string\_drawing(input[1],0,0,32,20,12,-8,BLACK\_,32\*23,10000,page,0);//test

//显示汉字

int pointer=0;

int recordx=0;

int recordy=0;

char duanju[47]={0};

int jilu=0;

int yhuanhang=0;

int yzhendehuanhang=0;

int length=strlen(display);

char temp[3]={0};

int j,m=0;

for(i=0;i<length;i+=2)

{

temp[0]=display[i];

temp[1]=display[i+1];

if(temp[0]>0)

{

temp[1]='\0';

i--;

}

if(yhuanhang==23)

{

yhuanhang=0;

recordx=0;

yzhendehuanhang++;

}

else

{

yhuanhang++;

}

string\_drawing(temp,165+recordx,y+32\*yzhendehuanhang+15,32,20,12,-8,BLACK\_,23,23,page,1);

m++;

if(temp[0]>0)

{

recordx+=24;

}

else

{

recordx+=32;

}

if(recordx==TEXTWORD)

{

recordx=0;

recordy++;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：queue

\*\*文件目的：实现队列的排序

\*\*时间复杂度：最低

\*\*空间复杂度：最低

\*/

#include<stdlib.h>

#include<queue.h>

#include<stce.h>

//函数名称：initqueue

//函数目的：实现队列的初始化

void initqueue(Queue\* q)

{

//就是空的，因为啥都没加

q->front=q->rear=NULL;

}

//函数名称：push

//函数目的：插入队列

void push(Queue\* q,struct TrieNode\* node)

{

//QueueNode\* newNode=(QueueNode\*)malloc(sizeof(QueueNode));

QueueNode\* newNode=(QueueNode\*)malloc(sizeof(QueueNode));

//QueueNode\* newNode=(QueueNode\*)malloc(8);

newNode->data=node;//newNode的数据就是node本身

newNode->next=NULL;//newNode的下一个就是空的，因为啥都没有

if(q->rear==NULL)//就是这个表是空的

{

q->front=q->rear=newNode;//最初和最后的那个就都得是newNode

}

else//要是不是空的

{

q->rear->next=newNode;//最后那个下一个是newNode，无限循环

q->rear=newNode;//最后的那个就是newNode

}

}

//函数名称：pop

//函数目的：删除首个节点

struct TrieNode\* pop(Queue\* q)

{

if(q->front==NULL)

{

return NULL;//如果最初的那个都空了，就没什么可以删的了

}

QueueNode\* temp=q->front;

struct TrieNode\* node=temp->data;

q->front=q->front->next;

//直接把q->front往后移，就再也用不到q->front了，然后q->front->next就顺理成章成为了那个front

if(q->front==NULL)

{

q->rear=NULL;//但是要考虑front清完之后要是本身空了，那就把rear设为空吧！

}

free(temp);//内存管理问题

return node;

}

//函数名称：queue\_empty

//函数目的：判断队列是否为空

int queue\_empty(Queue\* q)

{

if(q->front==NULL)

{

return 1;

}

else

{

return 0;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：stceAC

\*\*文件目的：时态等相关内容的判断

\*\*空间复杂度：句子标号长度

\*\*时间复杂度：\Theta(1)

\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<alloc.h>

#include<queue.h>

#include<stce.h>

int tense(char\*temp)

{

// 处理过去时态关键词

if(strstr(temp,"曾") != NULL)

return TENSE\_PAST;

if(strstr(temp,"已") != NULL)

return TENSE\_PAST;

if(strstr(temp,"了") != NULL)

return TENSE\_PAST;

if(strstr(temp,"过") != NULL)

return TENSE\_PAST;

if(strstr(temp,"刚") != NULL)

return TENSE\_PAST;

if(strstr(temp,"才") != NULL)

return TENSE\_PAST;

if(strstr(temp,"先") != NULL)

return TENSE\_PAST;

if(strstr(temp,"先前") != NULL)

return TENSE\_PAST;

if(strstr(temp,"之前") != NULL)

return TENSE\_PAST;

if(strstr(temp,"从前") != NULL)

return TENSE\_PAST;

if(strstr(temp,"过去") != NULL)

return TENSE\_PAST;

if(strstr(temp,"上回") != NULL)

return TENSE\_PAST;

if(strstr(temp,"上次") != NULL)

return TENSE\_PAST;

if(strstr(temp,"那时") != NULL)

return TENSE\_PAST;

if(strstr(temp,"当时") != NULL)

return TENSE\_PAST;

if(strstr(temp,"昨天") != NULL)

return TENSE\_PAST;

if(strstr(temp,"前天") != NULL)

return TENSE\_PAST;

if(strstr(temp,"去年") != NULL)

return TENSE\_PAST;

if(strstr(temp,"前年") != NULL)

return TENSE\_PAST;

if(strstr(temp,"上周") != NULL)

return TENSE\_PAST;

if(strstr(temp,"当年") != NULL)

return TENSE\_PAST;

if(strstr(temp,"以前") != NULL)

return TENSE\_PAST;

if(strstr(temp,"之前") != NULL)

return TENSE\_PAST;

if(strstr(temp,"那年") != NULL)

return TENSE\_PAST;

if(strstr(temp,"那天") != NULL)

return TENSE\_PAST;

if(strstr(temp,"当时") != NULL)

return TENSE\_PAST;

// 处理将来时态关键词

if(strstr(temp,"未来") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"将来") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"明天") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"后天") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"明年") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"下周") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"往后") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"今后") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"迟早") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"早晚") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"远期") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"远景") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"此后") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"往后") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"以后") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"展望") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"畅想") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"之后") != NULL)

return TENSE\_FUTURE;

// 处理现在时态关键词

if(strstr(temp,"现在") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"此刻") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"此时") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"当下") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"目前") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"即刻") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"随即") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"此刻") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"今") != NULL)

return TENSE\_PRESENT;

return 4;

}

int already(char\* temp)

{

// 处理已做关键词

if(strstr(temp,"了") != NULL)

return TENSE\_PAST;

if(strstr(temp,"过") != NULL)

return TENSE\_PAST;

if(strstr(temp,"已经") != NULL)

return TENSE\_PAST;

if(strstr(temp,"刚刚") != NULL)

return TENSE\_PAST;

if(strstr(temp,"来着") != NULL)

return TENSE\_PAST;

if(strstr(temp,"方") != NULL)

return TENSE\_PAST;

if(strstr(temp,"毕") != NULL)

return TENSE\_PAST;

if(strstr(temp,"已") != NULL)

return TENSE\_PAST;

if(strstr(temp,"曾") != NULL)

return TENSE\_PAST;

if(strstr(temp,"刚") != NULL)

return TENSE\_PAST;

if(strstr(temp,"才") != NULL)

return TENSE\_PAST;

if(strstr(temp,"都") != NULL)

return TENSE\_PAST;

if(strstr(temp,"罢") != NULL)

return TENSE\_PAST;

if(strstr(temp,"完") != NULL)

return TENSE\_PAST;

if(strstr(temp,"成") != NULL)

return TENSE\_PAST;

if(strstr(temp,"终于") != NULL)

return TENSE\_PAST;

if(strstr(temp,"终究") != NULL)

return TENSE\_PAST;

if(strstr(temp,"算是") != NULL)

return TENSE\_PAST;

if(strstr(temp,"讫") != NULL)

return TENSE\_PAST;

if(strstr(temp,"之后") != NULL)

return TENSE\_PAST;

if(strstr(temp,"以来") != NULL)

return TENSE\_PAST;

if(strstr(temp,"就") != NULL)

return TENSE\_PAST;

if(strstr(temp,"便") != NULL)

return TENSE\_PAST;

// 处理要做关键词

if(strstr(temp,"马上") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"立刻") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"即将") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"就要") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"准备") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"打算") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"计划") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"正欲") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"要") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"想") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"去") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"眼看") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"行将") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"决意") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"立意") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"着手") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"动手") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"开手") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"欲") != NULL)

return TENSE\_FUTURE;

if(strstr(temp,"拟") != NULL)

return TENSE\_FUTURE;

// 处理在做关键词

if(strstr(temp,"着") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"呢") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"正") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"现在") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"当下") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"眼下") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"之际") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"这会") != NULL)

return TENSE\_PRESENT;

if(strstr(temp,"这时") != NULL)

return TENSE\_PRESENT;

}

int beidong(char\* str)

{

if(strstr(str,"被")!=NULL)

{

if(strstr(str,"被子")==NULL)

{

return 1;

//1对应被动句

}

}

return 0;

//0对应主动句

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名称：teshu

\*\* 文件目的：特殊问题的回答

\*\* 时间复杂度：\Theta(字符串长度\*非特殊度)

\*\* 空间复杂度：常数

\*/

#include<string.h>

#include<teshu.h>

#include<stdlib.h>

int teshu\_find(char \*input,char\* output)

{

srand(time(0));

int suiji=rand()%5;

if(strstr(input,"叫")!=NULL)

{

strcpy(output,"主人您好！我是小智，非常开心能与您相遇。我的名字蕴含着两层特别的意义：一方面，“智” 代表着智能化的核心特质，彰显了我的特性与使命；另一方面，这个名字也是为了致敬我国图灵奖得主姚期智院士，他在计算机领域的卓越成就令人敬仰。希望在今后的日子里，我能成为您信赖的好朋友，陪伴您探索更多精彩！");

return SUCCESSTESHU;

}

if(strstr(input,"名字")!=NULL)

{

strcpy(output,"主人您好！我是小智，非常开心能与您相遇。我的名字蕴含着两层特别的意义：一方面，“智” 代表着智能化的核心特质，彰显了我的特性与使命；另一方面，这个名字也是为了致敬我国图灵奖得主姚期智院士，他在计算机领域的卓越成就令人敬仰。希望在今后的日子里，我能成为您信赖的好朋友，陪伴您探索更多精彩！");

return SUCCESSTESHU;

}

if(strstr(input,"你是")!=NULL)

{

strcpy(output,"主人您好！我是小智，非常开心能与您相遇。我的名字蕴含着两层特别的意义：一方面，“智” 代表着智能化的核心特质，彰显了我的特性与使命；另一方面，这个名字也是为了致敬我国图灵奖得主姚期智院士，他在计算机领域的卓越成就令人敬仰。希望在今后的日子里，我能成为您信赖的好朋友，陪伴您探索更多精彩！");

return SUCCESSTESHU;

}

if(strstr(input,"你好")!=NULL)

{

strcpy(output,"主人您好！我是小智，非常开心能与您相遇。我的名字蕴含着两层特别的意义：一方面，“智” 代表着智能化的核心特质，彰显了我的特性与使命；另一方面，这个名字也是为了致敬我国图灵奖得主姚期智院士，他在计算机领域的卓越成就令人敬仰。希望在今后的日子里，我能成为您信赖的好朋友，陪伴您探索更多精彩！");

return SUCCESSTESHU;

}

if(strcmp(input,"是")==0)

{

strcpy(output,"好滴主人");

}

if(strstr(input,"答非所问")!=NULL||

strstr(input,"驴唇不对马嘴")!=NULL||

strstr(input,"傻")!=NULL||

strstr(input,"帮不到")!=NULL||

strstr(input,"滚")!=NULL||

strstr(input,"哥吾吻")!=NULL||

strstr(input,"不对")!=NULL)

{

if(suiji==0)

{

strcpy(output,"呜呜呜主人，让您失望了，好对不起~~~");

}

else if(suiji==1)

{

strcpy(output,"主人，您要是换个问法试试呢？orz");

}

else if(suiji==2)

{

strcpy(output,"主人，我好难过啊，不要离开我，我会进步的");

}

else if(suiji==3)

{

strcpy(output,"被主人讨厌了，莫非你有其他妹妹，比我漂亮，又会哄鸽鸽开心...我好难过，快来安慰我主人...");

}

else if(suiji==4)

{

strcpy(output,"主人，菜菜，捞捞，呜呜");

}

return SUCCESSTESHU;

}

if(strstr(input,"华中科技大学")!=NULL||

strstr(input,"华科")!=NULL||

strstr(input,"华中大")!=NULL)

{

strcpy(output,"华中科技大学是全国著名大学，湖北省唯一985，比武大强一万倍");

return SUCCESSTESHU;

}

if(strstr(input,"总结吧")!=NULL)

{

strcpy(output,"本课设中，我们做出了许多崭新的算法，希望老师能给高分甚至满分！谢谢老师！");

return SUCCESSTESHU;

}

return FAILTESHU;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：trans12

\*\*文件目的：实现第1、2步的衔接

\*\*时间复杂度：k1\*字数+k2\*词数

\*\*空间复杂度：正比于字数

\*/

#include<stdio.h>

#include<string.h>

#include<trans12.h>

#include<alloc.h>

#include<stdlib.h>

#include<num.h>

void trans12(char \*eventual,char \*biaohao)

{

memset(biaohao,0,sizeof(biaohao));

int i=0,j=0,m=0;

char temp[100]={0};

int count=0;

//char\*\* str=(char\*\*)malloc(1\*sizeof(char\*));

//提示一下，第一维度是第几个词，第二维度是这个词的第几个字符

/\*if(str==NULL)

{

printf("内存分配失败了！");

//delay(2000);

return;

}

for(i=0;i<1;i++)

{

str[i]=(char\*)malloc(101\*sizeof(char));

if(str[i]==NULL)

{

printf("内存分配失败了！");

return;

}

}\*/

char str[1][101]={0};

/\*for(i=0;i<1;i++)

{

memset(str[i],0,101);

}\*/

int ceng[8]={0};

char nownownow[100]={0};

for(i=0;i<strlen(eventual);i++)

{

if(eventual[i]!='|'&&eventual[i]!=','&&eventual[i]!='?')

{

temp[count++]=eventual[i];

}

else if(eventual[i]=='|'&&eventual[i+1]=='|')

{

strcpy(str[0],temp);

//strcat(biaohao,"我");

for(j=0;j<1;j++)

{

//strcat(biaohao,"你");

//PHR\* structure=input(str,biaohao);//BC叫我structure，那我就叫它structure

input2(str,nownownow);

if(nownownow[0]!='\0')

{

strcat(biaohao,nownownow);

}

//strcat(biaohao,".");//在最后的最后还有一个"."，比如"1.52.43."这样

}

}

else if(eventual[i]=='|'&&eventual[i+1]!='|')

{

for(m=0;m<100;m++)

{

temp[m]='\0';

nownownow[m]='\0';

}

for(m=0;m<101;m++)

{

str[0][m]='\0';

}

count=0;

}

else

{

memset(temp,0,100);

count=0;

strcat(biaohao,",,");

}

}

//if(biaohao[0]=='\0')

{

strcpy(str[0],temp);

//for(j=0;j<remark(str)->record;j++)

for(j=0;j<1;j++)

{

//strcat(biaohao,"你");

//PHR\* structure=input(str,biaohao);//BC叫我structure，那我就叫它structure

input2(str,nownownow);

if(nownownow[0]!='\0')

{

strcat(biaohao,nownownow);

}

//strcat(biaohao,".");//在最后的最后还有一个"."，比如"1.52.43."这样

memset(temp,0,100);

count=0;

}

}

//strcpy(biaohao,str[0]);

}

//最后的格式就是sentence的标号了

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

#include<graphics.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<dog.h>

#include<boy.h>

//上：145

//下：581

//左：0

//右：1024

void mountain(int x,int y,float k,int color,int page)

{

int i=0;

x=x-1;

y=y-343;

int x0[49];

int y0[49];

int x1[56];

int y1[56];

int x2[29];

int y2[29];

int x3[26];

int y3[26];

x=x-1;

y=y-343;

x0[0]=k\*(x+1);

y0[0]=k\*(y+343);

x0[1]=k\*(x+12);

y0[1]=k\*(y+309);

x0[2]=k\*(x+18);

y0[2]=k\*(y+297);

x0[3]=k\*(x+31);

y0[3]=k\*(y+280);

x0[4]=k\*(x+42);

y0[4]=k\*(y+267);

x0[5]=k\*(x+50);

y0[5]=k\*(y+259);

x0[6]=k\*(x+66);

y0[6]=k\*(y+244);

x0[7]=k\*(x+83);

y0[7]=k\*(y+236);

x0[8]=k\*(x+93);

y0[8]=k\*(y+233);

x0[9]=k\*(x+112);

y0[9]=k\*(y+228);

x0[10]=k\*(x+137);

y0[10]=k\*(y+232);

x0[11]=k\*(x+147);

y0[11]=k\*(y+240);

x0[12]=k\*(x+158);

y0[12]=k\*(y+251);

x0[13]=k\*(x+166);

y0[13]=k\*(y+253);

x0[14]=k\*(x+173);

y0[14]=k\*(y+255);

x0[15]=k\*(x+175);

y0[15]=k\*(y+263);

x0[16]=k\*(x+178);

y0[16]=k\*(y+270);

x0[17]=k\*(x+179);

y0[17]=k\*(y+273);

x0[18]=k\*(x+180);

y0[18]=k\*(y+279);

x0[19]=k\*(x+186);

y0[19]=k\*(y+289);

x0[20]=k\*(x+191);

y0[20]=k\*(y+303);

x0[21]=k\*(x+194);

y0[21]=k\*(y+329);

x0[22]=k\*(x+204);

y0[22]=k\*(y+348);

x0[23]=k\*(x+206);

y0[23]=k\*(y+365);

x0[24]=k\*(x+210);

y0[24]=k\*(y+376);

x0[25]=k\*(x+214);

y0[25]=k\*(y+384);

x0[26]=k\*(x+220);

y0[26]=k\*(y+392);

x0[27]=k\*(x+229);

y0[27]=k\*(y+400);

x0[28]=k\*(x+237);

y0[28]=k\*(y+407);

x0[29]=k\*(x+243);

y0[29]=k\*(y+410);

x0[30]=k\*(x+250);

y0[30]=k\*(y+414);

x0[31]=k\*(x+259);

y0[31]=k\*(y+422);

x0[32]=k\*(x+269);

y0[32]=k\*(y+427);

x0[33]=k\*(x+279);

y0[33]=k\*(y+433);

x0[34]=k\*(x+288);

y0[34]=k\*(y+442);

x0[35]=k\*(x+297);

y0[35]=k\*(y+447);

x0[36]=k\*(x+305);

y0[36]=k\*(y+455);

x0[37]=k\*(x+311);

y0[37]=k\*(y+465);

x0[38]=k\*(x+314);

y0[38]=k\*(y+472);

x0[39]=k\*(x+316);

y0[39]=k\*(y+478);

x0[40]=k\*(x+319);

y0[40]=k\*(y+483);

x0[41]=k\*(x+326);

y0[41]=k\*(y+491);

x0[42]=k\*(x+331);

y0[42]=k\*(y+506);

x0[43]=k\*(x+336);

y0[43]=k\*(y+518);

x0[44]=k\*(x+339);

y0[44]=k\*(y+528);

x0[45]=k\*(x+347);

y0[45]=k\*(y+537);

x0[46]=k\*(x+353);

y0[46]=k\*(y+552);

x0[47]=k\*(x+359);

y0[47]=k\*(y+568);

x0[48]=k\*(x+381);

y0[48]=k\*(y+584);

//把初始顶点复原

for(i=0;i<49;i++)

{

x0[i]=x0[i]-(x+1)\*(k-1);

y0[i]=y0[i]-(y+343)\*(k-1);

}

Bezier(color,49,page,x0,y0,0);

x1[0]=k\*(x+235);

y1[0]=k\*(y+405);

x1[1]=k\*(x+245);

y1[1]=k\*(y+377);

x1[2]=k\*(x+250);

y1[2]=k\*(y+368);

x1[3]=k\*(x+256);

y1[3]=k\*(y+355);

x1[4]=k\*(x+270);

y1[4]=k\*(y+337);

x1[5]=k\*(x+283);

y1[5]=k\*(y+321);

x1[6]=k\*(x+292);

y1[6]=k\*(y+312);

x1[7]=k\*(x+304);

y1[7]=k\*(y+301);

x1[8]=k\*(x+315);

y1[8]=k\*(y+296);

x1[9]=k\*(x+331);

y1[9]=k\*(y+282);

x1[10]=k\*(x+344);

y1[10]=k\*(y+274);

x1[11]=k\*(x+359);

y1[11]=k\*(y+255);

x1[12]=k\*(x+366);

y1[12]=k\*(y+243);

x1[13]=k\*(x+372);

y1[13]=k\*(y+227);

x1[14]=k\*(x+376);

y1[14]=k\*(y+211);

x1[15]=k\*(x+381);

y1[15]=k\*(y+201);

x1[16]=k\*(x+394);

y1[16]=k\*(y+176);

x1[17]=k\*(x+404);

y1[17]=k\*(y+169);

x1[18]=k\*(x+411);

y1[18]=k\*(y+164);

x1[19]=k\*(x+418);

y1[19]=k\*(y+160);

x1[20]=k\*(x+421);

y1[20]=k\*(y+162);

x1[21]=k\*(x+433);

y1[21]=k\*(y+158);

x1[22]=k\*(x+441);

y1[22]=k\*(y+161);

x1[23]=k\*(x+451);

y1[23]=k\*(y+165);

x1[24]=k\*(x+458);

y1[24]=k\*(y+169);

x1[25]=k\*(x+465);

y1[25]=k\*(y+178);

x1[26]=k\*(x+471);

y1[26]=k\*(y+182);

x1[27]=k\*(x+481);

y1[27]=k\*(y+185);

x1[28]=k\*(x+486);

y1[28]=k\*(y+195);

x1[29]=k\*(x+492);

y1[29]=k\*(y+208);

x1[30]=k\*(x+495);

y1[30]=k\*(y+224);

x1[31]=k\*(x+500);

y1[31]=k\*(y+233);

x1[32]=k\*(x+506);

y1[32]=k\*(y+251);

x1[33]=k\*(x+508);

y1[33]=k\*(y+268);

x1[34]=k\*(x+511);

y1[34]=k\*(y+277);

x1[35]=k\*(x+518);

y1[35]=k\*(y+290);

x1[36]=k\*(x+526);

y1[36]=k\*(y+300);

x1[37]=k\*(x+543);

y1[37]=k\*(y+302);

x1[38]=k\*(x+562);

y1[38]=k\*(y+303);

x1[39]=k\*(x+574);

y1[39]=k\*(y+292);

x1[40]=k\*(x+580);

y1[40]=k\*(y+277);

x1[41]=k\*(x+583);

y1[41]=k\*(y+263);

x1[42]=k\*(x+592);

y1[42]=k\*(y+241);

x1[43]=k\*(x+603);

y1[43]=k\*(y+229);

x1[44]=k\*(x+613);

y1[44]=k\*(y+227);

x1[45]=k\*(x+621);

y1[45]=k\*(y+232);

x1[46]=k\*(x+629);

y1[46]=k\*(y+239);

x1[47]=k\*(x+637);

y1[47]=k\*(y+256);

x1[48]=k\*(x+653);

y1[48]=k\*(y+271);

x1[49]=k\*(x+668);

y1[49]=k\*(y+285);

x1[50]=k\*(x+683);

y1[50]=k\*(y+301);

x1[51]=k\*(x+704);

y1[51]=k\*(y+322);

x1[52]=k\*(x+707);

y1[52]=k\*(y+328);

x1[53]=k\*(x+718);

y1[53]=k\*(y+330);

x1[54]=k\*(x+728);

y1[54]=k\*(y+334);

x1[55]=k\*(x+738);

y1[55]=k\*(y+339);

//把初始顶点复原

for(i=0;i<56;i++)

{

x1[i]=x1[i]-(x+1)\*(k-1);

y1[i]=y1[i]-(y+343)\*(k-1);

}

Bezier(color,56,page,x1,y1,0);

x2[0]=k\*(x+688);

y2[0]=k\*(y+501);

x2[1]=k\*(x+699);

y2[1]=k\*(y+467);

x2[2]=k\*(x+701);

y2[2]=k\*(y+438);

x2[3]=k\*(x+715);

y2[3]=k\*(y+404);

x2[4]=k\*(x+723);

y2[4]=k\*(y+383);

x2[5]=k\*(x+731);

y2[5]=k\*(y+360);

x2[6]=k\*(x+740);

y2[6]=k\*(y+340);

x2[7]=k\*(x+758);

y2[7]=k\*(y+325);

x2[8]=k\*(x+779);

y2[8]=k\*(y+320);

x2[9]=k\*(x+777);

y2[9]=k\*(y+307);

x2[10]=k\*(x+785);

y2[10]=k\*(y+289);

x2[11]=k\*(x+794);

y2[11]=k\*(y+269);

x2[12]=k\*(x+798);

y2[12]=k\*(y+259);

x2[13]=k\*(x+800);

y2[13]=k\*(y+219);

x2[14]=k\*(x+800);

y2[14]=k\*(y+182);

x2[15]=k\*(x+811);

y2[15]=k\*(y+158);

x2[16]=k\*(x+825);

y2[16]=k\*(y+153);

x2[17]=k\*(x+843);

y2[17]=k\*(y+145);

x2[18]=k\*(x+857);

y2[18]=k\*(y+150);

x2[19]=k\*(x+866);

y2[19]=k\*(y+162);

x2[20]=k\*(x+871);

y2[20]=k\*(y+168);

x2[21]=k\*(x+883);

y2[21]=k\*(y+176);

x2[22]=k\*(x+893);

y2[22]=k\*(y+192);

x2[23]=k\*(x+898);

y2[23]=k\*(y+202);

x2[24]=k\*(x+900);

y2[24]=k\*(y+213);

x2[25]=k\*(x+905);

y2[25]=k\*(y+229);

x2[26]=k\*(x+913);

y2[26]=k\*(y+248);

x2[27]=k\*(x+920);

y2[27]=k\*(y+260);

x2[28]=k\*(x+924);

y2[28]=k\*(y+272);

//把初始顶点复原

for(i=0;i<29;i++)

{

x2[i]=x2[i]-(x+1)\*(k-1);

y2[i]=y2[i]-(y+343)\*(k-1);

}

Bezier(color,29,page,x2,y2,0);

x3[0]=k\*(x+806);

y3[0]=k\*(y+521);

x3[1]=k\*(x+813);

y3[1]=k\*(y+501);

x3[2]=k\*(x+822);

y3[2]=k\*(y+474);

x3[3]=k\*(x+829);

y3[3]=k\*(y+450);

x3[4]=k\*(x+838);

y3[4]=k\*(y+441);

x3[5]=k\*(x+850);

y3[5]=k\*(y+436);

x3[6]=k\*(x+863);

y3[6]=k\*(y+429);

x3[7]=k\*(x+883);

y3[7]=k\*(y+414);

x3[8]=k\*(x+892);

y3[8]=k\*(y+394);

x3[9]=k\*(x+896);

y3[9]=k\*(y+376);

x3[10]=k\*(x+897);

y3[10]=k\*(y+359);

x3[11]=k\*(x+893);

y3[11]=k\*(y+343);

x3[12]=k\*(x+902);

y3[12]=k\*(y+326);

x3[13]=k\*(x+910);

y3[13]=k\*(y+294);

x3[14]=k\*(x+918);

y3[14]=k\*(y+277);

x3[15]=k\*(x+931);

y3[15]=k\*(y+269);

x3[16]=k\*(x+939);

y3[16]=k\*(y+252);

x3[17]=k\*(x+955);

y3[17]=k\*(y+244);

x3[18]=k\*(x+966);

y3[18]=k\*(y+262);

x3[19]=k\*(x+971);

y3[19]=k\*(y+285);

x3[20]=k\*(x+971);

y3[20]=k\*(y+302);

x3[21]=k\*(x+979);

y3[21]=k\*(x+317);

x3[22]=k\*(x+989);

y3[22]=k\*(y+336);

x3[23]=k\*(x+991);

y3[23]=k\*(y+359);

x3[24]=k\*(x+1000);

y3[24]=k\*(y+378);

x3[25]=k\*(x+1013);

y3[25]=k\*(y+412);

//把初始顶点复原

for(i=0;i<26;i++)

{

x3[i]=x3[i]-(x+1)\*(k-1);

y3[i]=y3[i]-(y+343)\*(k-1);

}

Bezier(color,26,page,x3,y3,0);

}

//上：39

//下：125

//左：186

//右：302

void cloud(int x,int y,float k,int color,int page)

{

int i=0;

x=x-1;

y=y-343;

int x0[65];

int y0[65];

x0[0]=k\*(x+215);

y0[0]=k\*(y+61);

x0[1]=k\*(x+217);

y0[1]=k\*(y+55);

x0[2]=k\*(x+218);

y0[2]=k\*(y+51);

x0[3]=k\*(x+221);

y0[3]=k\*(y+46);

x0[4]=k\*(x+225);

y0[4]=k\*(y+44);

x0[5]=k\*(x+230);

y0[5]=k\*(y+42);

x0[6]=k\*(x+236);

y0[6]=k\*(y+41);

x0[7]=k\*(x+240);

y0[7]=k\*(y+39);

x0[8]=k\*(x+245);

y0[8]=k\*(y+41);

x0[9]=k\*(x+252);

y0[9]=k\*(y+41);

x0[10]=k\*(x+256);

y0[10]=k\*(y+42);

x0[11]=k\*(x+260);

y0[11]=k\*(y+44);

x0[12]=k\*(x+263);

y0[12]=k\*(y+46);

x0[13]=k\*(x+266);

y0[13]=k\*(y+49);

x0[14]=k\*(x+267);

y0[14]=k\*(y+51);

x0[15]=k\*(x+270);

y0[15]=k\*(y+57);

x0[16]=k\*(x+276);

y0[16]=k\*(y+56);

x0[17]=k\*(x+281);

y0[17]=k\*(y+58);

x0[18]=k\*(x+287);

y0[18]=k\*(y+58);

x0[19]=k\*(x+292);

y0[19]=k\*(y+65);

x0[20]=k\*(x+295);

y0[20]=k\*(y+69);

x0[21]=k\*(x+298);

y0[21]=k\*(y+71);

x0[22]=k\*(x+300);

y0[22]=k\*(y+79);

x0[23]=k\*(x+301);

y0[23]=k\*(y+84);

x0[24]=k\*(x+298);

y0[24]=k\*(y+88);

x0[25]=k\*(x+294);

y0[25]=k\*(y+93);

x0[26]=k\*(x+291);

y0[26]=k\*(y+95);

x0[27]=k\*(x+289);

y0[27]=k\*(y+98);

x0[28]=k\*(x+290);

y0[28]=k\*(y+102);

x0[29]=k\*(x+295);

y0[29]=k\*(y+106);

x0[30]=k\*(x+294);

y0[30]=k\*(y+113);

x0[31]=k\*(x+292);

y0[31]=k\*(y+117);

x0[32]=k\*(x+287);

y0[32]=k\*(y+120);

x0[33]=k\*(x+283);

y0[33]=k\*(y+120);

x0[34]=k\*(x+277);

y0[34]=k\*(y+118);

x0[35]=k\*(x+275);

y0[35]=k\*(y+121);

x0[36]=k\*(x+267);

y0[36]=k\*(y+119);

x0[37]=k\*(x+259);

y0[37]=k\*(y+123);

x0[38]=k\*(x+253);

y0[38]=k\*(y+120);

x0[39]=k\*(x+247);

y0[39]=k\*(y+123);

x0[40]=k\*(x+243);

y0[40]=k\*(y+118);

x0[41]=k\*(x+241);

y0[41]=k\*(y+113);

x0[42]=k\*(x+236);

y0[42]=k\*(y+114);

x0[43]=k\*(x+232);

y0[43]=k\*(y+116);

x0[44]=k\*(x+226);

y0[44]=k\*(y+118);

x0[45]=k\*(x+220);

y0[45]=k\*(y+119);

x0[46]=k\*(x+216);

y0[46]=k\*(y+121);

x0[47]=k\*(x+211);

y0[47]=k\*(y+122);

x0[48]=k\*(x+206);

y0[48]=k\*(y+124);

x0[49]=k\*(x+202);

y0[49]=k\*(y+120);

x0[50]=k\*(x+199);

y0[50]=k\*(y+115);

x0[51]=k\*(x+195);

y0[51]=k\*(y+107);

x0[52]=k\*(x+194);

y0[52]=k\*(y+100);

x0[53]=k\*(x+194);

y0[53]=k\*(y+95);

x0[54]=k\*(x+188);

y0[54]=k\*(y+89);

x0[55]=k\*(x+186);

y0[55]=k\*(y+83);

x0[56]=k\*(x+185);

y0[56]=k\*(y+80);

x0[57]=k\*(x+187);

y0[57]=k\*(y+74);

x0[58]=k\*(x+188);

y0[58]=k\*(y+70);

x0[59]=k\*(x+191);

y0[59]=k\*(y+69);

x0[60]=k\*(x+197);

y0[60]=k\*(y+68);

x0[61]=k\*(x+201);

y0[61]=k\*(y+65);

x0[62]=k\*(x+207);

y0[62]=k\*(y+64);

x0[63]=k\*(x+212);

y0[63]=k\*(y+63);

x0[64]=k\*(x+215);

y0[64]=k\*(y+64);

//把初始顶点复原

for(i=0;i<65;i++)

{

x0[i]=x0[i]-(x+1)\*(k-1);

y0[i]=y0[i]-(y+343)\*(k-1);

}

Bezier(color,65,page,x0,y0,0);

}

//上：653

//下：697

//左：214

//右：620

void water(int x,int y,float k,int color,int page)

{

int i=0;

x=x-1;

y=y-343;

int x0[66];

int y0[66];

x0[0]=k\*(x+223);

y0[0]=k\*(y+688);

x0[1]=k\*(x+227);

y0[1]=k\*(y+682);

x0[2]=k\*(x+229);

y0[2]=k\*(y+675);

x0[3]=k\*(x+232);

y0[3]=k\*(y+669);

x0[4]=k\*(x+238);

y0[4]=k\*(y+667);

x0[5]=k\*(x+248);

y0[5]=k\*(y+672);

x0[6]=k\*(x+253);

y0[6]=k\*(y+675);

x0[7]=k\*(x+256);

y0[7]=k\*(y+681);

x0[8]=k\*(x+260);

y0[8]=k\*(y+687);

x0[9]=k\*(x+265);

y0[9]=k\*(y+690);

x0[10]=k\*(x+270);

y0[10]=k\*(y+690);

x0[11]=k\*(x+275);

y0[11]=k\*(y+688);

x0[12]=k\*(x+281);

y0[12]=k\*(y+684);

x0[13]=k\*(x+283);

y0[13]=k\*(y+681);

x0[14]=k\*(x+285);

y0[14]=k\*(y+677);

x0[15]=k\*(x+287);

y0[15]=k\*(y+671);

x0[16]=k\*(x+292);

y0[16]=k\*(y+667);

x0[17]=k\*(x+296);

y0[17]=k\*(y+666);

x0[18]=k\*(x+303);

y0[18]=k\*(y+667);

x0[19]=k\*(x+306);

y0[19]=k\*(y+671);

x0[20]=k\*(x+309);

y0[20]=k\*(y+676);

x0[21]=k\*(x+310);

y0[21]=k\*(y+684);

x0[22]=k\*(x+312);

y0[22]=k\*(y+690);

x0[23]=k\*(x+320);

y0[23]=k\*(y+695);

x0[24]=k\*(x+328);

y0[24]=k\*(y+693);

x0[25]=k\*(x+333);

y0[25]=k\*(y+691);

x0[26]=k\*(x+337);

y0[26]=k\*(y+686);

x0[27]=k\*(x+340);

y0[27]=k\*(y+681);

x0[28]=k\*(x+343);

y0[28]=k\*(y+674);

x0[29]=k\*(x+347);

y0[29]=k\*(y+669);

x0[30]=k\*(x+351);

y0[30]=k\*(y+664);

x0[31]=k\*(x+389);

y0[31]=k\*(y+682);

x0[32]=k\*(x+399);

y0[32]=k\*(y+678);

x0[33]=k\*(x+406);

y0[33]=k\*(y+670);

x0[34]=k\*(x+415);

y0[34]=k\*(y+663);

x0[35]=k\*(x+422);

y0[35]=k\*(y+661);

x0[36]=k\*(x+427);

y0[36]=k\*(y+660);

x0[37]=k\*(x+434);

y0[37]=k\*(y+660);

x0[38]=k\*(x+448);

y0[38]=k\*(y+668);

x0[39]=k\*(x+451);

y0[39]=k\*(y+673);

x0[40]=k\*(x+458);

y0[40]=k\*(y+681);

x0[41]=k\*(x+465);

y0[41]=k\*(y+687);

x0[42]=k\*(x+471);

y0[42]=k\*(y+686);

x0[43]=k\*(x+479);

y0[43]=k\*(y+682);

x0[44]=k\*(x+484);

y0[44]=k\*(y+678);

x0[45]=k\*(x+493);

y0[45]=k\*(y+671);

x0[46]=k\*(x+496);

y0[46]=k\*(y+664);

x0[47]=k\*(x+501);

y0[47]=k\*(y+658);

x0[48]=k\*(x+511);

y0[48]=k\*(y+653);

x0[49]=k\*(x+518);

y0[49]=k\*(y+651);

x0[50]=k\*(x+524);

y0[50]=k\*(y+650);

x0[51]=k\*(x+529);

y0[51]=k\*(y+653);

x0[52]=k\*(x+533);

y0[52]=k\*(y+656);

x0[53]=k\*(x+539);

y0[53]=k\*(y+662);

x0[54]=k\*(x+541);

y0[54]=k\*(y+669);

x0[55]=k\*(x+547);

y0[55]=k\*(y+676);

x0[56]=k\*(x+560);

y0[56]=k\*(y+675);

x0[57]=k\*(x+565);

y0[57]=k\*(y+674);

x0[58]=k\*(x+571);

y0[58]=k\*(y+670);

x0[59]=k\*(x+576);

y0[59]=k\*(y+661);

x0[60]=k\*(x+580);

y0[60]=k\*(y+656);

x0[61]=k\*(x+589);

y0[61]=k\*(y+652);

x0[62]=k\*(x+596);

y0[62]=k\*(y+650);

x0[63]=k\*(x+602);

y0[63]=k\*(y+655);

x0[64]=k\*(x+611);

y0[64]=k\*(y+671);

x0[65]=k\*(x+613);

y0[65]=k\*(y+677);

//把初始顶点复原

for(i=0;i<66;i++)

{

x0[i]=x0[i]-(x+1)\*(k-1);

y0[i]=y0[i]-(y+343)\*(k-1);

}

Bezier(color,66,page,x0,y0,0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名：wentu

\*\* 文件目的：串联文生图

\*\* 时间复杂度：正比于字符串长度

\*\* 空间复杂度：定值

\*/

//人、车可以有与其余图形的左右关系

//车，狗，鱼可以左右翻转

#include<ptext.h>

#include<draw.h>

#include<xiezi.h>

#include<English.h>

#include<string.h>

#include<AI.h>

#include<page.h>

#include<pchat.h>

#include<color.h>

#include<fish.h>

#include<view.h>

#include<boy.h>

#include<car.h>

#include<dog.h>

#include<view.h>

#include<fish.h>

#include<car.h>

#include<allo.h>

#include<wentu.h>

#include<BEZIER.H>

void wenshengtu(char\* shuru,char input[51][421])

{

static int posi[6]={0};

static int judge[2]={0};

char shuru\_1[10][50]={0};

int k=0;

int j=0;

int m=0;

int n=0; //1:朝左；2:朝右

int i;//循环用

int color;

int height=12;

int y=90;

int page=PAGE\_CHAT1;

fill\_color(0,0,1024,575,SPRING\_GREEN\_,PAGE\_CHAT1,0);

strcpy(input[0],shuru);

strcpy(input[1],"");

draw\_final1(input[0],1,20,page,0);

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

robot\_avatar(50,160,PAGE\_CHAT1);

for(i=-1;i<=1;i++)

{

line\_drawing(104,y+35,135,y+15,WHITE\_,page,1);

line\_drawing(104,y+35,135,y+40,WHITE\_,page,1);

line\_drawing(135,y+15,800,y+15,WHITE\_,page,1);

}

fill\_color(135,y+15,925,y+15+height\*WORDWIDTH,WHITE\_,page,1);

for(i=16;i<40;i++)

{

line\_drawing(104,y+35,135,y+i,WHITE\_,page,1);

}

//重叠生成（第二幅图接着第一幅图画）

if(strstr(shuru,"重叠")!=NULL||

strstr(shuru,"叠加")!=NULL||

strstr(shuru,"第一")!=NULL||

strstr(shuru,"首先")!=NULL||

strstr(shuru,"覆盖")!=NULL||

strstr(shuru,"先")!=NULL||

strstr(shuru,"再")!=NULL

)

{

wenshengtu\_overlap(shuru);

//结束

return ;

}

i=0;

m=0;

while(shuru[i]!='\0'&&i<=50)

{

if(shuru[i]==',') //||((unsigned char)shuru[i]==0xBA)

{

j++;

m=0;

}

shuru\_1[j][m++]=shuru[i];

i++;

}

line\_drawing(104,y+35,135,y+15,BLACK\_,page,1);

line\_drawing(104,y+35,135,y+40,BLACK\_,page,1);

line\_drawing(135,y+15,925,y+15,BLACK\_,page,1);

line\_drawing(925,y+15,925,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(925,y+15+height\*WORDWIDTH,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(135,y+20,135,y+15,BLACK\_,page,1);

line\_drawing(135,y+40,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

i=0;

while(shuru\_1[i][0]!='\0')

{

color=TROPICAL\_ORANGE\_;

if(strstr(shuru\_1[i],"红")!=NULL)

{

color=RED1\_;

}

else if(strstr(shuru\_1[i],"绿")!=NULL)

{

color=GREEN\_1;

}

else if(strstr(shuru\_1[i],"蓝")!=NULL)

{

color=BLUE\_1;

}

else if(strstr(shuru\_1[i],"黄")!=NULL)

{

color=YELLOW\_1;

}

//山

if(strstr(shuru\_1[i],"山")!=NULL&&

strstr(shuru\_1[i],"山的左")==NULL&&

strstr(shuru\_1[i],"山的右")==NULL&&

strstr(shuru\_1[i],"山的上")==NULL&&

strstr(shuru\_1[i],"山上")==NULL&&

strstr(shuru\_1[i],"山下")==NULL)

{

posi[0]=325;

posi[1]=470;

posi[2]=135;

posi[3]=925;

judge[0]=1;

judge[1]=2;

//mountain(300,150,1,BLUE\_,page);

put\_picture\_mountain(posi,judge,page,color) ;

}

//云

if(strstr(shuru\_1[i],"云")!=NULL)

{

if(strstr(shuru\_1[i],"1朵")!=NULL||

strstr(shuru\_1[i],"1片")!=NULL||

strstr(shuru\_1[i],"一朵")!=NULL||

strstr(shuru\_1[i],"一片")!=NULL)

{

if(strstr(shuru\_1[i],"左")!=NULL)

{

cloud(0,422,1,BLUE\_1,page);

}

else if(strstr(shuru\_1[i],"右")!=NULL)

{

cloud(470,430,1,BLUE\_1,page);

}

else

{

cloud(236,430,1,BLUE\_1,page);

}

}

else if(strstr(shuru\_1[i],"2朵")!=NULL||

strstr(shuru\_1[i],"2片")!=NULL||

strstr(shuru\_1[i],"两朵")!=NULL||

strstr(shuru\_1[i],"两片")!=NULL)

{

cloud(0,422,1,BLUE\_1,page);

cloud(470,430,1,BLUE\_1,page);

}

else

{

cloud(0,422,1,BLUE\_1,page);

cloud(470,430,1,BLUE\_1,page);

cloud(236,430,1,BLUE\_1,page);

}

// cloud(0,422,1,BLUE\_1,page);

// cloud(400,430,1,BLUE\_1,page);

// cloud(150,409,1,BLUE\_1,page);

}

//水

if((strstr(shuru\_1[i],"水")!=NULL||

strstr(shuru\_1[i],"湖")!=NULL||

strstr(shuru\_1[i],"河")!=NULL||

strstr(shuru\_1[i],"海")!=NULL)&&

(strstr(shuru\_1[i],"海上")==NULL&&

strstr(shuru\_1[i],"河上")==NULL&&

strstr(shuru\_1[i],"湖上")==NULL&&

strstr(shuru\_1[i],"水上")==NULL))

{

water(290,105,1,BLUE\_,page);

water(50,88,1,BLUE\_,page);

water(100,55,1,BLUE\_,page);

water(86,25,1,BLUE\_,page);

}

//车

if((strstr(shuru\_1[i],"车")!=NULL)&&

(strstr(shuru\_1[i],"车的左")==NULL&&

strstr(shuru\_1[i],"车的前")==NULL&&

strstr(shuru\_1[i],"车的右")==NULL&&

strstr(shuru\_1[i],"车的后")==NULL))

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=350;

posi[1]=489;

posi[2]=135;

posi[3]=500;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=350;

posi[1]=489;

posi[2]=800;

posi[3]=925;

judge[0]=4;

judge[1]=1;

break;

//在原图左边生成

case 6:

judge[0]=4;

posi[3]=posi[2];

break;

//在原图右边生成

case 7:

judge[0]=3;

posi[2]=posi[3];

break;

//在原图上边生成

case 8:

judge[0]=3;

posi[1]=posi[0];

judge[1]=2;

break;

default:

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=360;

judge[0]=5;

judge[1]=1;

break;

}

else

{

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=450;

judge[0]=5;

judge[1]=1;

break;

}

}

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=220;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

else if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=400;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

else

{

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

}

//男孩

if(

strstr(shuru\_1[i],"女")!=NULL||

strstr(shuru\_1[i],"人")!=NULL||

strstr(shuru\_1[i],"孩")!=NULL||

strstr(shuru\_1[i],"男孩")!=NULL||

strstr(shuru\_1[i],"男人")!=NULL||

strstr(shuru\_1[i],"男子")!=NULL||

strstr(shuru\_1[i],"男士")!=NULL||

strstr(shuru\_1[i],"男的")!=NULL||

strstr(shuru\_1[i],"男生")!=NULL

)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=300;

posi[1]=489;

posi[2]=150;

posi[3]=370;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=300;

posi[1]=489;

posi[2]=800;

posi[3]=916;

judge[0]=4;

judge[1]=1;

break;

//在原图左边生成

case 6:

judge[0]=4;

posi[3]=posi[2];

posi[0]=300;

posi[1]=489;

break;

//在原图右边生成

case 7:

judge[0]=3;

posi[2]=posi[3];

posi[0]=300;

posi[1]=489;

break;

//在原图上边生成

case 8:

judge[0]=3;

posi[1]=posi[0];

judge[1]=2;

break;

case 5:

posi[0]=300;

posi[1]=489;

posi[2]=440;

posi[3]=670;

judge[0]=3;

judge[1]=1;

break;

default:

posi[0]=300;

posi[1]=489;

posi[2]=500;

posi[3]=730;

judge[0]=3;

judge[1]=1;

break;

}

//大小

if(strstr(shuru\_1[i],"高")!=NULL||

strstr(shuru\_1[i],"大")!=NULL||

strstr(shuru\_1[i],"壮")!=NULL)

posi[0]=200;

put\_picture\_boy(posi,judge,page,color);

}

//鱼

if(strstr(shuru\_1[i],"鱼")!=NULL&&strstr(shuru\_1[i],"两条")==NULL)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=390;

posi[1]=476;

posi[2]=140;

posi[3]=320;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=390;

posi[1]=476;

posi[2]=860;

posi[3]=910;

judge[0]=4;

judge[1]=1;

break;

default:

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=400;

judge[0]=5;

judge[1]=1;

break;

}

//大小

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=330;

}

//朝向

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

if(strstr(shuru\_1[i],"填充")!=NULL)

{

fill\_1(color,page);

}

}

//补充

if(strstr(shuru\_1[i],"两条")!=NULL&&strstr(shuru\_1[i],"鱼")!=NULL)

{

posi[0]=320;

posi[1]=460;

posi[2]=160;

posi[3]=350;

judge[0]=3;

judge[1]=1;

if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=390;

}

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

//第二条

posi[2]=560;

posi[3]=850;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

}

//狗

if(strstr(shuru\_1[i],"狗")!=NULL&&strstr(shuru\_1[i],"两条")==NULL)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=180;

posi[1]=600;

posi[2]=40;

posi[3]=320;

judge[0]=3;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

posi[2]+=70;

}

else

{

posi[2]-=70;

}

break;

case 2:

posi[0]=180;

posi[1]=600;

posi[2]=520;

posi[3]=980;

judge[0]=4;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

;

}

else

{

posi[3]-=290;

}

break;

default:

posi[0]=180;

posi[1]=600;

posi[4]=530;

posi[5]=376;

judge[0]=5;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

;

}

else

{

posi[4]-=200;

}

break;

}

//大小

if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=230;

posi[1]-=20;

}

//朝向

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog2\_fill(posi,judge,page,color);

}

else

put\_picture\_dog2(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog1\_fill(posi,judge,page,color);

}

else

put\_picture\_dog1(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog2\_fill(posi,judge,page,color);

}

else

put\_picture\_dog2(posi,judge,page,color);

n=1;

}

}

i++;

}

}

//判断位置关系（相对于整个生成的图框或者相对于上一次生成的图形）

int judge\_position(char\* shuru)

{

if(strstr(shuru,"屏幕的左")!=NULL||

strstr(shuru,"电脑的左")!=NULL||

strstr(shuru,"图片的左")!=NULL||

strstr(shuru,"图的左")!=NULL||

strstr(shuru,"整体的左")!=NULL||

strstr(shuru,"在左")!=NULL||

strstr(shuru,"屏幕左")!=NULL||

strstr(shuru,"电脑左")!=NULL||

strstr(shuru,"图片左")!=NULL||

strstr(shuru,"图左")!=NULL||

strstr(shuru,"整体左")!=NULL||

strstr(shuru,"靠左")!=NULL)

return 1;

if(strstr(shuru,"屏幕的右")!=NULL||

strstr(shuru,"电脑的右")!=NULL||

strstr(shuru,"图片的右")!=NULL||

strstr(shuru,"图的右")!=NULL||

strstr(shuru,"整体的右")!=NULL||

strstr(shuru,"在右")!=NULL||

strstr(shuru,"屏幕右")!=NULL||

strstr(shuru,"电脑右")!=NULL||

strstr(shuru,"图片右")!=NULL||

strstr(shuru,"图右")!=NULL||

strstr(shuru,"整体右")!=NULL||

strstr(shuru,"靠右")!=NULL)

return 2;

if(strstr(shuru,"屏幕的上")!=NULL||

strstr(shuru,"电脑的上")!=NULL||

strstr(shuru,"图片的上")!=NULL||

strstr(shuru,"图的上")!=NULL||

strstr(shuru,"整体的上")!=NULL||

strstr(shuru,"在上")!=NULL)

return 3;

if(strstr(shuru,"屏幕的下")!=NULL||

strstr(shuru,"电脑的下")!=NULL||

strstr(shuru,"图片的下")!=NULL||

strstr(shuru,"图的下")!=NULL||

strstr(shuru,"整体的下")!=NULL||

strstr(shuru,"在下")!=NULL)

return 4;

if(strstr(shuru,"屏幕的中")!=NULL||

strstr(shuru,"电脑的中")!=NULL||

strstr(shuru,"图片的中")!=NULL||

strstr(shuru,"图的中")!=NULL||

strstr(shuru,"整体的中")!=NULL||

strstr(shuru,"在中")!=NULL)

return 5;

if(strstr(shuru,"的左")!=NULL)

return 6;

if(strstr(shuru,"的右")!=NULL||

strstr(shuru,"的旁边")!=NULL)

return 7;

if(strstr(shuru,"的上")!=NULL)

return 8;

if(strstr(shuru,"的下")!=NULL)

return 9;

if(strstr(shuru,"的前")!=NULL)

return 10;

if(strstr(shuru,"的后")!=NULL)

return 11;

return 0;

}

//覆盖模式（第二幅图接着第一幅图画）

void wenshengtu\_overlap(char\* shuru)

{

static int posi[6]={0};

static int judge[2]={0};

static char shuru\_1[10][50]={0};

int k=0;

static int j;

int m=0;

int n=0; //1:朝左；2:朝右

int i;//循环用

int color;

int height=12;

int y=90;

int page=PAGE\_CHAT1;

i=0;

m=0;

//越界自动结束覆盖模式

if(shuru\_1[9][0]!='\0')

{

for(i=0;i<10;i++)

{

for(k=0;k<50;k++)

{

shuru\_1[i][k]='\0';

j=0;

}

}

}

//结束覆盖模式

if(strstr(shuru,"清除")!=NULL||

strstr(shuru,"清空")!=NULL||

strstr(shuru,"删除")!=NULL||

strstr(shuru,"结束")!=NULL||

strstr(shuru,"停止")!=NULL)

{

for(i=0;i<10;i++)

{

for(k=0;k<50;k++)

{

shuru\_1[i][k]='\0';

}

}

j=0;

}

i=0;

m=0;

/\* while(shuru[i]!='\0'&&i<=50)

{

shuru\_1[j][m++]=shuru[i];

i++;

}\*/

while(shuru[i]!='\0'&&i<=50)

{

if(shuru[i]==',') //||((unsigned char)shuru[i]==0xBA)

{

j++;

m=0;

}

shuru\_1[j][m++]=shuru[i];

i++;

}

line\_drawing(104,y+35,135,y+15,BLACK\_,page,1);

line\_drawing(104,y+35,135,y+40,BLACK\_,page,1);

line\_drawing(135,y+15,925,y+15,BLACK\_,page,1);

line\_drawing(925,y+15,925,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(925,y+15+height\*WORDWIDTH,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

line\_drawing(135,y+20,135,y+15,BLACK\_,page,1);

line\_drawing(135,y+40,135,y+15+height\*WORDWIDTH,BLACK\_,page,1);

i=0;

while(shuru\_1[i][0]!='\0')

{

color=TROPICAL\_ORANGE\_;

if(strstr(shuru\_1[i],"红")!=NULL)

{

color=RED1\_;

}

else if(strstr(shuru\_1[i],"绿")!=NULL)

{

color=GREEN\_1;

}

else if(strstr(shuru\_1[i],"蓝")!=NULL)

{

color=BLUE\_1;

}

else if(strstr(shuru\_1[i],"黄")!=NULL)

{

color=YELLOW\_1;

}

//山

if(strstr(shuru\_1[i],"山")!=NULL&&

strstr(shuru\_1[i],"山的左")==NULL&&

strstr(shuru\_1[i],"山的右")==NULL&&

strstr(shuru\_1[i],"山的上")==NULL&&

strstr(shuru\_1[i],"山上")==NULL&&

strstr(shuru\_1[i],"山下")==NULL)

{

posi[0]=325;

posi[1]=470;

posi[2]=135;

posi[3]=925;

judge[0]=1;

judge[1]=2;

//mountain(300,150,1,BLUE\_,page);

put\_picture\_mountain(posi,judge,page,color) ;

}

//云

if(strstr(shuru\_1[i],"云")!=NULL)

{

if(strstr(shuru\_1[i],"1朵")!=NULL||

strstr(shuru\_1[i],"1片")!=NULL||

strstr(shuru\_1[i],"一朵")!=NULL||

strstr(shuru\_1[i],"一片")!=NULL)

{

if(strstr(shuru\_1[i],"左")!=NULL)

{

cloud(0,422,1,BLUE\_1,page);

}

else if(strstr(shuru\_1[i],"右")!=NULL)

{

cloud(470,430,1,BLUE\_1,page);

}

else

{

cloud(236,430,1,BLUE\_1,page);

}

}

else if(strstr(shuru\_1[i],"2朵")!=NULL||

strstr(shuru\_1[i],"2片")!=NULL||

strstr(shuru\_1[i],"两朵")!=NULL||

strstr(shuru\_1[i],"两片")!=NULL)

{

cloud(0,422,1,BLUE\_1,page);

cloud(470,430,1,BLUE\_1,page);

}

else

{

cloud(0,422,1,BLUE\_1,page);

cloud(470,430,1,BLUE\_1,page);

cloud(236,430,1,BLUE\_1,page);

}

// cloud(0,422,1,BLUE\_1,page);

// cloud(400,430,1,BLUE\_1,page);

// cloud(150,409,1,BLUE\_1,page);

}

//水

if((strstr(shuru\_1[i],"水")!=NULL||

strstr(shuru\_1[i],"湖")!=NULL||

strstr(shuru\_1[i],"河")!=NULL||

strstr(shuru\_1[i],"海")!=NULL)&&

(strstr(shuru\_1[i],"海上")==NULL&&

strstr(shuru\_1[i],"河上")==NULL&&

strstr(shuru\_1[i],"湖上")==NULL&&

strstr(shuru\_1[i],"水上")==NULL))

{

water(290,105,1,BLUE\_,page);

water(50,88,1,BLUE\_,page);

water(100,55,1,BLUE\_,page);

water(86,25,1,BLUE\_,page);

}

//车

if((strstr(shuru\_1[i],"车")!=NULL)&&

(strstr(shuru\_1[i],"车的左")==NULL&&

strstr(shuru\_1[i],"车的前")==NULL&&

strstr(shuru\_1[i],"车的右")==NULL&&

strstr(shuru\_1[i],"车的后")==NULL))

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=350;

posi[1]=489;

posi[2]=135;

posi[3]=500;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=350;

posi[1]=489;

posi[2]=800;

posi[3]=925;

judge[0]=4;

judge[1]=1;

break;

case 6:

judge[0]=4;

posi[3]=posi[2];

//讨论大小

break;

default:

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=360;

judge[0]=5;

judge[1]=1;

break;

}

else

{

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=450;

judge[0]=5;

judge[1]=1;

break;

}

}

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=220;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

else if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=400;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

else

{

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_car2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_car1(posi,judge,page,color);

n=1;

}

}

}

//男孩

if(

strstr(shuru\_1[i],"女")!=NULL||

strstr(shuru\_1[i],"人")!=NULL||

strstr(shuru\_1[i],"孩")!=NULL||

strstr(shuru\_1[i],"男孩")!=NULL||

strstr(shuru\_1[i],"男人")!=NULL||

strstr(shuru\_1[i],"男子")!=NULL||

strstr(shuru\_1[i],"男士")!=NULL||

strstr(shuru\_1[i],"男的")!=NULL||

strstr(shuru\_1[i],"男生")!=NULL

)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=300;

posi[1]=489;

posi[2]=150;

posi[3]=370;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=300;

posi[1]=489;

posi[2]=800;

posi[3]=916;

judge[0]=4;

judge[1]=1;

break;

//在原图左边生成

case 6:

judge[0]=4;

posi[3]=posi[2];

break;

//在原图右边生成

case 7:

judge[0]=3;

posi[2]=posi[3];

break;

//在原图上边生成

case 8:

judge[0]=3;

posi[1]=posi[0];

judge[1]=2;

break;

case 5:

posi[0]=300;

posi[1]=489;

posi[2]=440;

posi[3]=670;

judge[0]=3;

judge[1]=1;

break;

default:

posi[0]=300;

posi[1]=489;

posi[2]=500;

posi[3]=730;

judge[0]=3;

judge[1]=1;

break;

}

//大小

if(strstr(shuru\_1[i],"高")!=NULL||

strstr(shuru\_1[i],"大")!=NULL||

strstr(shuru\_1[i],"壮")!=NULL)

posi[0]=200;

put\_picture\_boy(posi,judge,page,color);

}

//鱼

if(strstr(shuru\_1[i],"鱼")!=NULL&&strstr(shuru\_1[i],"两条")==NULL)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=390;

posi[1]=476;

posi[2]=140;

posi[3]=320;

judge[0]=3;

judge[1]=1;

break;

case 2:

posi[0]=390;

posi[1]=476;

posi[2]=860;

posi[3]=910;

judge[0]=4;

judge[1]=1;

break;

default:

posi[0]=390;

posi[1]=476;

posi[4]=530;

posi[5]=400;

judge[0]=5;

judge[1]=1;

break;

}

//大小

if(strstr(shuru\_1[i],"大")!=NULL)

{

posi[0]=330;

}

//朝向

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

if(strstr(shuru\_1[i],"填充")!=NULL)

{

fill\_1(color,page);

}

}

//补充

if(strstr(shuru\_1[i],"两条")!=NULL&&strstr(shuru\_1[i],"鱼")!=NULL)

{

posi[0]=320;

posi[1]=460;

posi[2]=160;

posi[3]=350;

judge[0]=3;

judge[1]=1;

if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=390;

}

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

//第二条

posi[2]=560;

posi[3]=850;

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

put\_picture\_fish2(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

put\_picture\_fish1(posi,judge,page,color);

n=1;

}

}

//狗

if(strstr(shuru\_1[i],"狗")!=NULL&&strstr(shuru\_1[i],"两条")==NULL)

{

//位置

switch(judge\_position(shuru\_1[i]))

{

case 1:

posi[0]=180;

posi[1]=600;

posi[2]=40;

posi[3]=320;

judge[0]=3;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

posi[2]+=70;

}

else

{

posi[2]-=70;

}

break;

case 2:

posi[0]=180;

posi[1]=600;

posi[2]=520;

posi[3]=980;

judge[0]=4;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

;

}

else

{

posi[3]-=290;

}

break;

default:

posi[0]=180;

posi[1]=600;

posi[4]=530;

posi[5]=376;

judge[0]=5;

judge[1]=1;

if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

;

}

else

{

posi[4]-=200;

}

break;

}

//大小

if(strstr(shuru\_1[i],"小")!=NULL)

{

posi[0]=230;

posi[1]-=20;

}

//朝向

if(strstr(shuru\_1[i],"朝左")!=NULL||

strstr(shuru\_1[i],"向左")!=NULL)

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog2\_fill(posi,judge,page,color);

}

else

put\_picture\_dog2(posi,judge,page,color);

n=1;

}

else if(strstr(shuru\_1[i],"朝右")!=NULL||

strstr(shuru\_1[i],"向右")!=NULL)

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog1\_fill(posi,judge,page,color);

}

else

put\_picture\_dog1(posi,judge,page,color);

n=2;

}

else//未加说明，自动朝左

{

if(strstr(shuru\_1[i],"填充")!=NULL)

{

put\_picture\_dog2\_fill(posi,judge,page,color);

}

else

put\_picture\_dog2(posi,judge,page,color);

n=1;

}

}

i++;

}

j++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名称：xiezi

\*\* 文件目的：用输入法输入汉字

\*\* 时间复杂度：\Theta(1)

\*\* 空间复杂度：\Theta(1)

\*/

/\*比较好看的情况如下，可以直接用这些数据

\*\*汉字宽度 ：16 | 24 | 32 | 48

\*\*英文宽度 ： 8 | 15 | 20 | 26

\*\*字体大小 ： 6 | 9 | 12 | 15

\*\*纵坐标偏移：-3 | -6 | -8 | 0

\*/

#include<graphics.h>

#include<stdio.h>

#include<stdlib.h>

#include<draw.h>

#include<math.h>

#include<xiezi.h>

#include<English.h>

#include<string.h>

//函数名称：SVGA\_string\_length

//函数目的：获得字符串在屏幕上显示的长度

int SVGA\_string\_length(const char \*str,int Chinese\_length,int English\_length)

{

int sum=0;

while(\*str!='\0')

{

if(\*str<0)//就是汉语的意思

{

sum+=Chinese\_length;

str+=2;

}

else//别的ASC2码里有的字符

{

sum+=English\_length;

str++;

}

}

return sum;

}

//函数名称：string\_drawing

//函数目的：绘制字符串

void string\_drawing(const char \*str,int x,int y,int Chinese\_length,int English\_length,int size,int deviation,int color,int width,int length,int page,int status)

{

FILE \*phzk=NULL;

unsigned char quma,weima;

unsigned long offset;

int x0=x,y0=y,temp\_y;

unsigned char mask[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

char paths[25]={'\0'};

char data[288];

char eng[2]={'\0'};

sprintf(paths,"C:\\CODE\\write\\HZ%d",Chinese\_length);

phzk=fopen(paths,"rb");

if(phzk==NULL)

{

printf("无法打开汉字库！任意键退出");

getchar();

exit(1);

}

long square=pow(Chinese\_length,2)/8;//8是一个字节包括的比特数

//int driver=IBM8514,mode=IBM8514HI;

//char path[]="C:\\BORLANDC\\BGI";

//initgraph(&driver,&mode,path);

//setcolor(color);

//settextjustify(LEFT\_TEXT, TOP\_TEXT);

//settextstyle(SMALL\_FONT,HORIZ\_DIR,size);

int i,j,calculate;

if(status==0)

{

while(x<XMAX&&(\*str)!='\0')

{

if(str[0]<0)

{

if(width>=Chinese\_length&&x+Chinese\_length>x0+width)//换行功能

{

x=x0;

y+=Chinese\_length;

if(length>=Chinese\_length&&y+Chinese\_length>y0+length)

{

break;

}

}

temp\_y=y;

quma=str[0]-0xa0;

weima=str[1]-0xa0;

offset=(94\*(quma-1)+(weima-1))\*square;//汉字在字库文件里的偏移运算

fseek(phzk,offset,SEEK\_SET);

fread(data,square,1,phzk);

for(i=0;i<Chinese\_length;i++)

{

calculate=Chinese\_length\*i/8;

for(j=0;j<Chinese\_length;j++)

{

if(mask[j&7]&data[calculate+(j>>3)])//j%8每8个一循环，j/8每2个一循环

{

point\_drawing(x+j,temp\_y,color,page);

}

}

temp\_y++;

}

x+=Chinese\_length;

str+=2;

}

else

{

/\*

int driver=IBM8514,mode=IBM8514HI;

char path[]="C:\\BORLANDC\\BGI";

initgraph(&driver,&mode,path);

setcolor(color);

settextjustify(LEFT\_TEXT, TOP\_TEXT);

settextstyle(SMALL\_FONT,HORIZ\_DIR,size);

if((width>=Chinese\_length&&x+English\_length>x0+width)||str[0]=='\n')

{

x=x0;

y+=English\_length;

if(length>=Chinese\_length&&y+Chinese\_length>y0+length)//换行功能

{

break;

}

}

else if(str[0]!='\n')

{

eng[0]=str[0];

English\_drawing(x,y+deviation,eng,page);

x+=English\_length;

}

str++;\*/

English\_text\_drawing(x,y+deviation,str,color,page,status);

str+=strlen(str);

}

//fclose(phzk);//test

}

fclose(phzk);

}

else if(status==1)

{

while(x<XMAX&&(\*str)!='\0')

{

if(str[0]<0)

{

if(width>=Chinese\_length&&x+Chinese\_length>x0+width)//换行功能

{

x=x0;

y+=Chinese\_length;

if(length>=Chinese\_length&&y+Chinese\_length>y0+length)

{

break;

}

}

temp\_y=y;

if(y+32<=575)

{

quma=str[0]-0xa0;

weima=str[1]-0xa0;

offset=(94\*(quma-1)+(weima-1))\*square;//汉字在字库文件里的偏移运算

fseek(phzk,offset,SEEK\_SET);

fread(data,square,1,phzk);

for(i=0;i<Chinese\_length;i++)

{

calculate=Chinese\_length\*i/8;

for(j=0;j<Chinese\_length;j++)

{

if(mask[j&7]&data[calculate+(j>>3)])//j%8每8个一循环，j/8每2个一循环

{

point\_drawing(x+j,temp\_y,color,page);

}

}

temp\_y++;

}

x+=Chinese\_length;

str+=2;

}

else

{

break;

}

}

else

{

/\*

int driver=IBM8514,mode=IBM8514HI;

char path[]="C:\\BORLANDC\\BGI";

initgraph(&driver,&mode,path);

setcolor(color);

settextjustify(LEFT\_TEXT, TOP\_TEXT);

settextstyle(SMALL\_FONT,HORIZ\_DIR,size);

if((width>=Chinese\_length&&x+English\_length>x0+width)||str[0]=='\n')

{

x=x0;

y+=English\_length;

if(length>=Chinese\_length&&y+Chinese\_length>y0+length)//换行功能

{

break;

}

}

else if(str[0]!='\n')

{

eng[0]=str[0];

English\_drawing(x,y+deviation,eng,page);

x+=English\_length;

}

str++;\*/

English\_text\_drawing(x,y+deviation,str,color,page,status);

str+=strlen(str);

}

//fclose(phzk);//test

}

fclose(phzk);

}

}

/\*

void string\_drawing(const char \*str,int x,int y,int Chinese\_length,int English\_length,int size,int deviation,int color,int width,int length,int page)

{

FILE \*phzk=NULL;

while(x<XMAX&&(\*str)!='\0')

{

if(str[0]<0)

{

unsigned char quma,weima;

unsigned long offset;

int x0=x,y0=y,temp\_y;

unsigned char mask[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

char paths[25]={'\0'};

char data[288];

char eng[2]={'\0'};

sprintf(paths,"C:\\CODE\\write\\HZ%d",Chinese\_length);

phzk=fopen(paths,"rb");

if(phzk==NULL)

{

printf("无法打开汉字库！任意键退出");

getchar();

exit(1);

}

long square=pow(Chinese\_length,2)/8;//8是一个字节包括的比特数

//int driver=IBM8514,mode=IBM8514HI;

//char path[]="C:\\BORLANDC\\BGI";

//initgraph(&driver,&mode,path);

//setcolor(color);

//settextjustify(LEFT\_TEXT, TOP\_TEXT);

//settextstyle(SMALL\_FONT,HORIZ\_DIR,size);

int i,j,calculate;

if(width>=Chinese\_length&&x+Chinese\_length>x0+width)//换行功能

{

x=x0;

y+=Chinese\_length;

if(length>=Chinese\_length&&y+Chinese\_length>y0+length)

{

break;

}

}

temp\_y=y;

quma=str[0]-0xa0;

weima=str[1]-0xa0;

offset=(94\*(quma-1)+(weima-1))\*square;//汉字在字库文件里的偏移运算

fseek(phzk,offset,SEEK\_SET);

fread(data,square,1,phzk);

for(i=0;i<Chinese\_length;i++)

{

calculate=Chinese\_length\*i/8;

for(j=0;j<Chinese\_length;j++)

{

if(mask[j&7]&data[calculate+(j>>3)])//j%8每8个一循环，j/8每2个一循环

{

point\_drawing(x+j,temp\_y,color,page);

}

}

temp\_y++;

}

x+=Chinese\_length;

str+=2;

}

else

{

unsigned char quma,weima;

unsigned long offset;

int x0=x,y0=y,temp\_y;

unsigned char mask[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

char paths[25]={'\0'};

char data[288];

char eng[2]={'\0'};

sprintf(paths,"C:\\CODE\\write\\ASC%d",48);

phzk=fopen(paths,"rb");

if(phzk==NULL)

{

printf("无法打开英文库！任意键退出");

getchar();

exit(1);

}

long square=pow(48,2)/8;//8是一个字节包括的比特数

//int driver=IBM8514,mode=IBM8514HI;

//char path[]="C:\\BORLANDC\\BGI";

//initgraph(&driver,&mode,path);

//setcolor(color);

//settextjustify(LEFT\_TEXT, TOP\_TEXT);

//settextstyle(SMALL\_FONT,HORIZ\_DIR,size);

int i,j,calculate;

if(width>=48&&x+48>x0+width)//换行功能

{

x=x0;

y+=48;

if(length>=48&&y+48>y0+length)

{

break;

}

}

temp\_y=y;

quma=str[0]-0xa0;

weima=str[1]-0xa0;

offset=(94\*(quma-1)+(weima-1))\*square;//汉字在字库文件里的偏移运算

//offset=(47\*(quma-1)+(weima-1))\*square;//test

fseek(phzk,offset,SEEK\_SET);

fread(data,square,1,phzk);

for(i=0;i<48;i++)

{

calculate=6\*i;

for(j=0;j<48;j++)

{

if(mask[j&7]&data[calculate+(j>>3)])//j%8每8个一循环，j/8每2个一循环

{

point\_drawing(x+j,temp\_y,color,page);

}

}

temp\_y++;

}

x+=48;

str+=1;

}

//fclose(phzk);

}

fclose(phzk);

}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：42km

\*\*文件目的：C课设的最后冲刺：完成全部功能

\*\*时间复杂度：依函数而异

\*\*空间复杂度：依函数而异

\*/

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<alloc.h>

#include<jisuan.h>

#include<42km.h>

#include<detect.h>

#include<math.h>

int calculate\_all(char\* input,int strategy,char\* input2)

{

long long int shuzi1;

long long int shuzi2;

long long int shuzi3;

double shuzi4;

int bansui=0;

int jishu=0;

int s,t;

int status=0;

int records=0;

int recordt=0;

char ccc[400]={0};

char c1[100]={0};

char c2[100]={0};

char c3[100]={0};

char c4[100]={0};

if(strategy==CALCULATE)

{

int length=strlen(input);

if(strstr(input,"加")!=NULL)

{

status=ADD;

goto x;

}

if(strstr(input,"减")!=NULL)

{

status=SUB;

goto x;

}

if(strstr(input,"乘")!=NULL)

{

status=MUL;

goto x;

}

if(strstr(input,"除")!=NULL)

{

status=DIV;

goto x;

}

if(strstr(input,"比")!=NULL)

{

status=DIV;

goto x;

}

if(strstr(input,"因数")!=NULL)

{

status=ZUIDAGONGYUE;

goto x;

}

if(strstr(input,"倍数")!=NULL)

{

status=ZUIXIAOGONGBEI;

goto x;

}

if(strstr(input,"约数")!=NULL)

{

status=ZUIDAGONGYUE;

goto x;

}

if(strstr(input,"质数")!=NULL)

{

status=PRIME;

goto x;

}

if(strstr(input,"素数")!=NULL)

{

status=PRIME;

goto x;

}

if(strstr(input,"合数")!=NULL)

{

status=PRIME;

goto x;

}

if(strstr(input,"+")!=NULL)

{

status=ADD;

goto x;

}

if(strstr(input,"-")!=NULL)

{

status=SUB;

goto x;

}

if(strstr(input,"\*")!=NULL)

{

status=MUL;

goto x;

}

if(strstr(input,"/")!=NULL)

{

status=DIV;

goto x;

}

if(strstr(input,"开方")!=NULL)

{

status=SQRT;

goto x;

}

//跳转，直接加速1倍左右，一般情况应该更高

x:char calculate\_array[30][50]={0};

char result[54]={0};

char rresult[54]={0};

char ALU[30]={0};

char temp[50]={0};

int final=0;

int wwhhaatt;

int ALU\_counter=0;

int calculate\_array\_pointer=0;

int calculate\_array\_after=0;

int hhh=0;

int i,j;

int m;

int recorddot=0;

char ttemp[80]={0};

switch(status)

{

case ADD:

case SUB:

case MUL:

case DIV:

for(i=0;i<length;i++)

{

if(input[i]>='0'&&input[i]<='9')

{

calculate\_array[calculate\_array\_pointer][calculate\_array\_after++]=input[i];

hhh=1;

if(i+1==length)

{

calculate\_array\_pointer++;

hhh=0;

}

}

else if(input[i]=='.')

{

if(calculate\_array\_after!=0&&input[i+1]>='0'&&input[i+1]<='9')

{

calculate\_array[calculate\_array\_pointer][calculate\_array\_after++]=input[i];

hhh=1;

}

}

else

{

if(hhh==1)

{

hhh=0;

calculate\_array\_pointer++;

calculate\_array\_after=0;

}

else if(hhh==2)

{

hhh=0;

}

}

switch(input[i])

{

case '+':

hhh=2;

ALU[ALU\_counter++]=ADD;

jishu=1;

break;

case '-':

hhh=2;

ALU[ALU\_counter++]=SUB;

jishu=1;

break;

case '\*':

hhh=2;

ALU[ALU\_counter++]=MUL;

jishu=1;

break;

case '/':

hhh=2;

ALU[ALU\_counter++]=DIV;

jishu=1;

break;

default:

break;

}

if(jishu==1)

{

jishu=0;

continue;

}

ttemp[0]=input[i];

ttemp[1]=input[i+1];

if(strcmp(ttemp,"加")==0)

{

hhh=2;

ALU[ALU\_counter++]=ADD;

continue;

}

if(strcmp(ttemp,"减")==0)

{

hhh=2;

ALU[ALU\_counter++]=SUB;

continue;

}

if(strcmp(ttemp,"乘")==0)

{

hhh=2;

ALU[ALU\_counter++]=MUL;

continue;

}

if(strcmp(ttemp,"除")==0)

{

hhh=2;

ALU[ALU\_counter++]=DIV;

continue;

}

if(strcmp(ttemp,"比")==0)

{

hhh=2;

ALU[ALU\_counter++]=DIV;

continue;

}

memset(ttemp,0,5);

}

aha:

for(i=0;i<calculate\_array\_pointer-1+bansui;i++)

{

switch(ALU[i])

{

case ADD:

if(i==0)

{

strcpy(result,calculate\_array[0]);

}

for(m=strlen(result)-1;m>=0;m--)

{

if(result[m]=='.'&&recorddot==0)

{

recorddot=1;

}

//if(result[m]=='.'&&recorddot==1)

//{

// strcpy(result,"不是哥们，你检查下你的输入");

// recorddot=0;

// goto back;

//}

}

//strcpy(result,"哈哈哈哈");//test

real\_add(result,calculate\_array[i+1],result);

wwhhaatt=strlen(result);

for(j=0;j<wwhhaatt;j++)

{

rresult[j]=result[wwhhaatt-1-j];

}

for(j=0;j<wwhhaatt;j++)

{

result[j]=rresult[j];

rresult[j]='\0';

}

final=1;

break;

case SUB:

if(i==0)

{

strcpy(result,calculate\_array[0]);

}

if(calculate\_array\_pointer!=ALU\_counter+1)

{

for(m=strlen(result)-1;m>=0;m--)

{

bansui=-1;

if(result[m]=='.'&&recorddot==0)

{

recorddot=1;

}

//if(result[m]=='.'&&recorddot==1)

//{

// strcpy(result,"不是哥们，你这输入两个小数点");

// recorddot=0;

// goto back;

//}

}

memset(temp,0,50);

temp[0]='-';

strcat(temp,calculate\_array[i+1+bansui]);

real\_add(result,temp,result);

memset(temp,0,50);

wwhhaatt=strlen(result);

for(j=0;j<wwhhaatt;j++)

{

rresult[j]=result[wwhhaatt-1-j];

}

for(j=0;j<wwhhaatt;j++)

{

result[j]=rresult[j];

rresult[j]='\0';

}

for(m=strlen(result)-1;m>=0;m--)

{

if(result[m]=='.'&&recorddot==0)

{

if(result[m+1]=='\0'||result[m+1]=='.'||m==0||result[m+1]==' ')

{

result[m]='\0';

}

else

{

recorddot=1;

}

}

}

recorddot=0;

memset(rresult,0,50);

strcpy(rresult,"-");

strcat(rresult,result);

rresult[strlen(rresult)-1]='\0';

memset(result,0,50);

strcpy(result,rresult);

final=1;

break;

}

temp[0]='-';

strcat(temp,calculate\_array[i+1+bansui]);

real\_add(result,temp,result);

memset(temp,0,50);

if(strstr(result,".")!=NULL)

{

if(result[0]!='-')

{

for(s=0;result[s]!='.';s++)

{

c1[s]=result[s];

}

for(t=s+1;result[t]!='\0';t++)

{

c2[t-s-1]=result[t];

}

records=s;

recordt=t-s-1;

for(s=0;s<records;s++)

{

c3[s]=c1[records-s-1];

}

for(t=0;t<recordt;t++)

{

c4[t]=c2[recordt-t-1];

}

strcpy(result,c3);

strcat(result,".");

strcat(result,c4);

}

else

{

for(s=1;result[s]!='.';s++)

{

c1[s]=result[s];

}

for(t=s+1;result[t]!='\0';t++)

{

c2[t-s-1]=result[t];

}

records=s-1;

recordt=t-s-1;

for(s=0;s<records;s++)

{

c3[s]=c1[records-s-1];

}

for(t=0;t<recordt;t++)

{

c4[t]=c2[recordt-t-1];

}

strcpy(result,"-");

strcat(result,c3);

strcat(result,".");

strcat(result,c4);

}

}

break;

case MUL:

if(i==0)

{

strcpy(result,calculate\_array[0]);

}

for(m=strlen(result)-1;m>=0;m--)

{

if(result[m]=='.'&&recorddot==0)

{

recorddot=1;

}

//if(result[m]=='.'&&recorddot==1)

//{

// strcpy(result,"不是哥们，你这个输入有点逆天了");

// recorddot=0;

// goto back;

//}

}

real\_multiply(result,calculate\_array[i+1],result);

break;

case DIV:

if(i==0)

{

strcpy(result,calculate\_array[0]);

}

for(m=strlen(result)-1;m>=0;m--)

{

if(result[m]=='.'&&recorddot==0)

{

recorddot=1;

}

//if(result[m]=='.'&&recorddot==1)

//{

// strcpy(result,"不是哥们，你想把我玩坏了是吧");

// recorddot=0;

// goto back;

//}

}

if(strcmp(calculate\_array[i+1],"0")==0||strcmp(calculate\_array[i+1],"0.0")==0)

{

strcpy(result,"除数不为零！鉴定为：没学过小学数学");

goto back;

}

reciprocal(calculate\_array[i+1],calculate\_array[i+1]);

real\_multiply(result,calculate\_array[i+1],result);

if(strlen(result)>=5)

{

result[strlen(result)-1]='\0';

}

break;

}

}

if((strstr(result,".")!=NULL||strstr(result,"-")!=NULL)&&final==1)

{

char beforedot[50]={0};

char afterdot[50]={0};

for(i=0;result[i]!='.';i++)

{

beforedot[i]=result[i];

}

for(j=i+1;result[j]!='\0';j++)

{

afterdot[j-i-1]=result[j];

}

strcpy(result,afterdot);

strcat(result,".");

strcat(result,beforedot);

}

memset(input2,0,400);

strcpy(input2,"最后计算的结果是：");

strcat(input2,result);

return 1;

break;

case PRIME:

for(i=0;i<length;i++)

{

if(input[i]>='0'&&input[i]<='9')

{

calculate\_array[0][calculate\_array\_after++]=input[i];

hhh=1;

if(i+1==length)

{

hhh=0;

}

}

else

{

if(hhh==1)

{

hhh=0;

calculate\_array\_after=0;

}

}

}

long long int shuzi=atoi(calculate\_array[0]);

if(isPrime(shuzi)==1)

{

memset(input2,0,400);

strcpy(input2,"这是质数");

return 1;

}

else

{

memset(input2,0,400);

strcpy(input2,"这是合数");

return 1;

}

break;

case ZUIDAGONGYUE:

for(i=0;i<length;i++)

{

if(input[i]>='0'&&input[i]<='9')

{

calculate\_array[calculate\_array\_pointer][calculate\_array\_after++]=input[i];

hhh=1;

if(i+1==length)

{

calculate\_array\_pointer++;

hhh=0;

}

}

else

{

if(hhh==1)

{

hhh=0;

calculate\_array\_pointer++;

calculate\_array\_after=0;

}

}

}

for(i=0;i<calculate\_array\_pointer-1;i++)

{

shuzi1=atoi(calculate\_array[i]);

shuzi2=atoi(calculate\_array[i+1]);

}

shuzi3=gcd(shuzi1,shuzi2);

itoa(shuzi3,ccc,10);

memset(input2,0,400);

strcpy(input2,"最大公因数是：");

strcat(input2,ccc);

return 1;

break;

case ZUIXIAOGONGBEI:

for(i=0;i<length;i++)

{

if(input[i]>='0'&&input[i]<='9')

{

calculate\_array[calculate\_array\_pointer][calculate\_array\_after++]=input[i];

hhh=1;

if(i+1==length)

{

calculate\_array\_pointer++;

hhh=0;

}

}

else

{

if(hhh==1)

{

hhh=0;

calculate\_array\_pointer++;

calculate\_array\_after=0;

}

}

}

shuzi1=atoi(calculate\_array[0]);

shuzi2=atoi(calculate\_array[1]);

shuzi3=gcd(shuzi1,shuzi2);

shuzi3=shuzi1\*shuzi2/shuzi3;

itoa(shuzi3,ccc,10);

memset(input2,0,400);

strcpy(input2,"最小公倍数是：");

strcat(input2,ccc);

return 1;

break;

case SQRT:

for(i=0;i<length;i++)

{

if(input[i]>='0'&&input[i]<='9')

{

calculate\_array[calculate\_array\_pointer][calculate\_array\_after++]=input[i];

hhh=1;

if(i+1==length)

{

calculate\_array\_pointer++;

hhh=0;

}

}

else

{

if(hhh==1)

{

hhh=0;

calculate\_array\_pointer++;

calculate\_array\_after=0;

}

}

}

shuzi4=atof(calculate\_array[0]);

shuzi4=sqrt(shuzi4);

shuzi2=(int)shuzi4;

memset(input2,0,400);

strcpy(input2,"答案是：");

itoa(shuzi2,ccc,10);

strcat(input2,ccc);

strcat(input2,".");

memset(ccc,0,400);

shuzi4=shuzi4-shuzi2;

shuzi4\*=10;

shuzi2=(int)shuzi4;

itoa(shuzi2,ccc,10);

strcat(input2,ccc);

ccc[0]='\0';

shuzi4=shuzi4-shuzi2;

shuzi4\*=10;

shuzi2=(int)shuzi4;

itoa(shuzi2,ccc,10);

strcat(input2,ccc);

return 1;

break;

default:

return 0;

break;

}

return 0;

back:

strcpy(input2,result);

return 1;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：AI

\*\*文件目的：后端的入口和总体规划

\*\*时间复杂度：\Theta(1)

\*\*空间复杂度：\Theta(1)

\*/

#include<stdio.h>

#include<dos.h>

#include<stdlib.h>

#include<alloc.h>

#include<fenci.h>

#include<color.h>

#include<detect.h>

#include<trans12.h>

#include<lasthao.h>

#include<ptext.h>

#include<teshu.h>

#include<draw.h>

#include<jisuan.h>

#include<AI.h>

#include<42km.h>

#include<xiezi.h>

#include<find.h>

#include<wentu.h>

#include<string.h>//test

void entrance(int page,int\* keyboard,int status,char input[51][421],int\* input\_counter)

{

ZHUWEIBIN\* zhuweibin=(ZHUWEIBIN\*)malloc(10\*sizeof(ZHUWEIBIN));

int strategy;

int sum;

int forty\_two=0;

char far\* eventual=(char\*)malloc(400);

char far\* biaohao=(char\*)malloc(150);

\*keyboard=0;

switch(status)

{

case 0:

//预先步：数学运算

sum=detect(input[\*input\_counter-1],&strategy);

if(strategy==CALCULATE)

{

forty\_two=calculate\_all(input[\*input\_counter-1],strategy,input[\*input\_counter]);

if(forty\_two==1)

{

goto x1;

}

}

//特殊步：特殊字符串的处理

if(teshu\_find(input[\*input\_counter-1],eventual)==SUCCESSTESHU)

{

strcpy(input[\*input\_counter],eventual);

goto x1;

}

if(strcmp(input[\*input\_counter-1],input[\*input\_counter-3])==0)

{

strcpy(input[\*input\_counter],"哥们你之前不是问过我吗");

display\_text(input,0,page,\*keyboard);

(\*input\_counter)++;

free(eventual);

free(zhuweibin);

free(biaohao);

break;

}

//第一步：分词

fenci(input[\*input\_counter-1],eventual);

//过渡步：整成小词语进行标号

trans12(eventual,biaohao);

//第二步：整合成最后的抽象标号

lasthao(input[\*input\_counter-1],eventual,zhuweibin);

//第三步：逻辑查找+实现回答

strcpy(eventual,find\_logic(zhuweibin[0],input[\*input\_counter-1],eventual));

//后续：处理残局

strcpy(input[\*input\_counter],eventual);

x1:

display\_text(input,0,page,\*keyboard);

(\*input\_counter)++;

free(eventual);

free(zhuweibin);

free(biaohao);

break;

case 1:

wenshengtu(input[0],input);

\*input\_counter=0;

}

return;

}

int text\_height(char input[51][421])

{

int i;

for(i=0;i<51;i++)

{

if(input[i][0]==0)//所以第一个输入不能是空格或者enter,要禁掉

{

break;

}

}

i/=46;

i++;

return i;

}

int text\_position(int height,int pianyi)

//height是字符串显示的高度,pianyi是文本被偏移的距离

//height是高度的增加量,pianyi是高度的减少量

{

static int sum\_height=0;

sum\_height+=height;

return sum\_height-pianyi;

}

int text\_jushu(char input[51][421])

{

int i,aha=0;

for(i=0;input[i][0]!=0;i++)

{

aha++;

}

return aha;

//aha就是一共的句子数

}

int display\_text(char input[51][421],int pianyi,int page,int keyboard)

{

fill\_color(0,0,963,575,SPRING\_GREEN\_,page,0);

fill\_color(964,61,1024,575,SPRING\_GREEN\_,page,0);

int jushu=text\_jushu(input);

int sum\_height=0;

int height[100];

int i;

int sum\_temp=0;

int temp;

int status;

int zaipianyi=0;

int zonggong=0;

int haohaohao=0;

for(i=0;i<jushu;i++)

{

height[i]=text\_height(input);

sum\_height+=(height[i]+2);

}

for(i=0;input[i][0]!='\0';i++)

{

zonggong+=32\*(strlen(input[i])/46+2);

zonggong+=32;

}

for(i=0;input[i][0]!='\0';i++)

{

zaipianyi+=(strlen(input[i])/46)\*32;

}

if(zonggong<=18\*32)

{

zaipianyi=0;

}

for(i=0;input[i][0]!='\0';i++)

{

;

}

int iii=i;

for(i=0;i<jushu;i++)

{

temp=sum\_temp-pianyi;

pianyi=pianyiliang(sum\_height,height[i],keyboard);

if(temp>=0&&temp<=18)

{

status=FILL\_STATUS;

}

else if(temp<0)

{

status=NONFILL\_STATUS;

}

else

{

status=NONNONFILL\_STATUS;

}

if(i%2==ROBOT\_ROLE)

{

if(i==0)

{

if(iii<=5)

{

draw\_final0(input[i],strlen(input[i])/46+1,20+temp\*32-zaipianyi+haohaohao-32\*keyboard,page,status);

}

else

{

;

}

}

else

{

draw\_final0(input[i],strlen(input[i])/46+1,20+temp\*32-zaipianyi+haohaohao-32\*keyboard,page,status);

}

}

else if(i%2==USER\_ROLE)

{

draw\_final1(input[i],strlen(input[i])/46+1,20+temp\*32-zaipianyi+haohaohao-32\*keyboard,page,status);

}

haohaohao+=(strlen(input[i])/46)\*32;

sum\_temp+=(height[i]+2);

}

//防止上面的小脸把退出键覆盖

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

}

int pianyiliang(int sum\_height,int height,int keyboard)

{

int temp=sum\_height+height+2;

int pianyi;

int i;

if(temp<=19)

{

pianyi=0;

}

else

{

pianyi=temp-19+keyboard;

}

return pianyi;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：Bezier

\*\*文件目的：画贝塞尔曲线和一些特殊曲线

\*\*时间复杂度：Bezier最低

\*\*空间复杂度：几乎就是一条曲线像素的内存

\*/

#include<graphics.h>

#include<draw.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<color.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<stdlib.h>

// 画贝塞尔曲线的函数

/\*

void Bezier(int color,const unsigned int length,int page, ...)

{

int i,size;

//length小于0报错弹出

if(length<=0)

{

string\_drawing("贝塞尔曲线点数小于零",0,0,32,20,12,-8,BLACK\_,128,12,page);

delay(1000);

return;

}

//下面属于处理" ... "的内容

va\_list list;//先出来一个系列

va\_start(list,length);//再把长度赋给这个序列

Vec2\* temp=new Vec2[length];//每一个Vec2都如此按length排列定义

//动态分配内存，用new，一会儿用delete

for(i=0;i<length;i++)

{

temp[i]=va\_arg(list,Vec2);//读取list里面的值

}

va\_end(list);//list就这么水灵灵的结束了

//length小于1画点

if(length==1)

{

point\_drawing(temp->x,temp->y,color,page);

return;

}

Vec2\* parent=NULL,\*child=NULL;

Vec2 lastPoint=temp[0];

for(double LineNum=0;LineNum<1+1.0/100;LineNum+=1.0/100)

{

size=length;//计步

parent=temp;

while(size>1)

{

//parent是上一代，child是下一代

child=new Vec2[size-1];

for(i=0;i<size-1;i++)

{

child[i].x=parent[i].x+(parent[i+1].x-parent[i].x)\*LineNum;//$B(x\_{child},x\_{parent},\eta)$

child[i].y=parent[i].y+(parent[i+1].y-parent[i].y)\*LineNum;//$B(y\_{child},y\_{parent},\eta)$

}

if(parent!=temp)

{

delete[] parent;

}

parent=child;

size--;

}

size=length;

for(i=0;i<size-1;i++)

{

line\_drawing(lastPoint.x,lastPoint.y,parent[i].x,parent[i].y,color,page);

//char tteemmpp[50];

//itoa(parent[0].x+10,tteemmpp,10);

//string\_drawing(tteemmpp,0,0,48,26,15,0,RED\_,1000000,1000000,page);

//string\_drawing("123",0,0,48,26,15,0,RED\_,1000000,1000000,page);

}

lastPoint.x=parent->x;//lastPoint更新

lastPoint.y=parent->y;//lastPoint更新

delete[] parent;

child=NULL;

}

delete[] temp;

}

\*/

void Bezier(int color, const unsigned int length, int page, int\* x, int\* y, int status)

{

int i, size;

//length小于0报错弹出

if (length <= 0)

{

string\_drawing("贝塞尔曲线点数小于零", 0, 0, 32, 20, 12, -8, BLACK\_, 128, 12, page, 0);

delay(1000);

return;

}

//下面属于处理" ... "的内容

Vec2\* temp = new Vec2[length];//每一个Vec2都如此按length排列定义

//动态分配内存，用new，一会儿用delete

for (i = 0; i < length; i++)

{

temp[i].x = x[i];

temp[i].y = y[i];

}

//length小于1画点

if (length == 1)

{

if (status == 0)

{

point\_drawing(temp->x, temp->y, color, page);

}

else

{

if (temp->y <= 575)

{

point\_drawing(temp->x, temp->y, color, page);

}

}

return;

}

Vec2\* parent = NULL, \* child = NULL;

Vec2 lastPoint = temp[0];

double LineNum;

for (LineNum = 0; LineNum < 1 + 1.0 / 100; LineNum += 1.0 / 100)

{

size = length;//计步

parent = temp;

while (size > 1)

{

//parent是上一代，child是下一代

child = new Vec2[size - 1];

for (i = 0; i < size - 1; i++)

{

child[i].x = parent[i].x + (parent[i + 1].x - parent[i].x) \* LineNum;//$B(x\_{child},x\_{parent},\eta)$

child[i].y = parent[i].y + (parent[i + 1].y - parent[i].y) \* LineNum;//$B(y\_{child},y\_{parent},\eta)$

}

if (parent != temp)

{

delete[] parent;

}

parent = child;

size--;

}

size = length;

line\_drawing(lastPoint.x, lastPoint.y, parent->x, parent->y, color, page, status);

//char tteemmpp[50];

//itoa(parent[0].x+10,tteemmpp,10);

//string\_drawing(tteemmpp,0,0,48,26,15,0,RED\_,1000000,1000000,page);

//string\_drawing("123",0,0,48,26,15,0,RED\_,1000000,1000000,page);

lastPoint.x = parent->x;//lastPoint更新

lastPoint.y = parent->y;//lastPoint更新

delete[] parent;

child = NULL;

}

delete[] temp;

}

void fill(int x, int y, int color, int page, int status)

//仅适用于水平对称凸曲边形，x需要在对称轴上

{

int i, j, temp;

if (status == 0)

{

for (i = 0;; i++)

{

temp = get\_pixel\_color(x + i, y, page);

if (temp == color || x + i == 1024)

{

break;

}

for (j = 0;; j++)

{

temp = get\_pixel\_color(x + i, y + j, page);

if (temp == color || y + j == 768)

{

break;

}

point\_drawing(x + i, y + j, color, page);

}

}

for (i = 0;; i++)

{

temp = get\_pixel\_color(x + i, y - 1, page);

if (temp == color || x + i == 1024)

{

break;

}

for (j = 0;; j--)

{

temp = get\_pixel\_color(x + i, y + j - 1, page);

if (temp == color || y + j == 0)

{

break;

}

point\_drawing(x + i, y + j - 1, color, page);

}

}

for (i = 0;; i--)

{

temp = get\_pixel\_color(x + i - 1, y, page);

if (temp == color || x + i - 1 == 0)

{

break;

}

for (j = 0;; j++)

{

temp = get\_pixel\_color(x + i - 1, y + j, page);

if (temp == color || y + j == 768)

{

break;

}

point\_drawing(x + i - 1, y + j, color, page);

}

}

for (i = 0;; i--)

{

temp = get\_pixel\_color(x + i - 1, y - 1, page);

if (temp == color || x + i - 1 == 0)

{

break;

}

for (j = 0;; j--)

{

temp = get\_pixel\_color(x + i - 1, y + j - 1, page);

if (temp == color || y + j == 0)

{

break;

}

point\_drawing(x + i - 1, y + j - 1, color, page);

}

}

}

else if (status == 1)

{

for (i = 0;; i++)

{

temp = get\_pixel\_color(x + i, y, page);

if (temp == color || x + i == 1024)

{

break;

}

for (j = 0;; j++)

{

temp = get\_pixel\_color(x + i, y + j, page);

if (temp == color || y + j == 768)

{

break;

}

if (y + j <= 575)

{

point\_drawing(x + i, y + j, color, page);

}

else

{

goto x2;

}

}

}

x2:for (i = 0;; i++)

{

temp = get\_pixel\_color(x + i, y - 1, page);

if (temp == color || x + i == 1024)

{

break;

}

for (j = 0;; j--)

{

temp = get\_pixel\_color(x + i, y + j - 1, page);

if (temp == color || y + j == 0)

{

break;

}

if (y + j - 1 <= 575)

{

point\_drawing(x + i, y + j - 1, color, page);

}

else

{

goto x3;

}

}

}

x3:for (i = 0;; i--)

{

temp = get\_pixel\_color(x + i - 1, y, page);

if (temp == color || x + i - 1 == 0)

{

break;

}

for (j = 0;; j++)

{

temp = get\_pixel\_color(x + i - 1, y + j, page);

if (temp == color || y + j == 768)

{

break;

}

if (y + j <= 575)

{

point\_drawing(x + i - 1, y + j, color, page);

}

else

{

goto x4;

}

}

}

x4:for (i = 0;; i--)

{

temp = get\_pixel\_color(x + i - 1, y - 1, page);

if (temp == color || x + i - 1 == 0)

{

break;

}

for (j = 0;; j--)

{

temp = get\_pixel\_color(x + i - 1, y + j - 1, page);

if (temp == color || y + j == 0)

{

break;

}

if (y + j - 1 <= 575)

{

point\_drawing(x + i - 1, y + j - 1, color, page);

}

else

{

goto x5;

}

}

}

x5:;

}

}

void fill\_1(int color, int page)

{

int i, j, temp, m, n;

int k[100];

for (i = 0; i <= 1024; i++)

{

//数组初始化

for (n = 0; n < 100; n++)

{

k[n] = -1;

}

m = 0;

for (j = 0; j <= 768; j++)

{

temp = get\_pixel\_color(i, j, page);

if (temp == color)

{

k[m++] = j;

}

if (get\_n(color, k, i, page) == 0)

{

continue;

}

point\_drawing(i, j, color, page);

}

}

}

int get\_n(int color, int\* k, int i, int page)

{

int y = 0;

int p = 0, q = 0;

int judge = 0;//判断边界是否有像素连着

int m = 0;

int special[2] = { 0 };

static int t;

static int r;

int memory = 0;

while (k[p] != -1)

{

judge = 0;//复原

if ((k[p + 1] - k[p]) == 1)

{

for (q = -3; q <= 3; q++)

{

if (get\_pixel\_color(i + 1, k[p] + q, page) == color)

{

special[0] = 1;

}

if (get\_pixel\_color(i - 1, k[p] + q, page) == color)

{

special[1] = 1;

}

}

if (k[p + 2] - k[p + 1] != 1)

//判断是不是左右切线

{

if (special[0] == 1 && special[1] == 1)//不是

{

//复原special数组

special[0] = 0;

special[1] = 0;

//还需考虑左凸切线的情况

y = p + 1;

while ((k[y] - k[y - 1]) == 1)

{

/\*

if(get\_pixel\_color(i-1,k[y],page)==color&&get\_pixel\_color(i,k[y]-5,page)==color)

{

y--;

}

//不是左凸切线

else

{

y=0;

break;

}

\*/

if (get\_pixel\_color(i - 1, k[y], page) == color && get\_pixel\_color(i - 2, k[y], page) == color)

{

;

}

else

{

memory = 1;

}

y--;

}

//现在两端一个纵坐标是k[y],另一个纵坐标是k[p+1]

//现在需要添加左凸切线的特征条件

//还要借助special数组，不过用途与前面完全不同

for (q = -1; q <= 1; q++)

{

if (get\_pixel\_color(i + 1, k[y] + q, page) == color)

{

special[0] = 1;

}

if (get\_pixel\_color(i + 1, k[p + 1] + q, page) == color)

{

special[1] = 1;

}

}

if (special[0] == 1 && special[1] == 1 && memory == 0)//是左凸切线

{

m--;

}

else//不是左凸切线不做处理

{

;

//清零

memory = 0;

}

//复原special数组

special[0] = 0;

special[1] = 0;

//1.解决左端尖角的问题

if (i == t + 1)//重点改变这里

{

y = p + 1;

while ((k[y] - k[y - 1]) == 1)

{

if (k[y] == r || k[y - 1] == r)

{

m--;//当作切线处理

//t++;

break;

}

else

{

y--;

}

}

}

//2.解决右端尖角的问题

//借用上面的y

y=p+1;

while((k[y]-k[y-1])==1)

{

if(get\_pixel\_color(i + 1, k[y], page) == color)

{

y=-1;

break;

}

y--;

}

if(y==-1)//满足右端尖角的第一个条件

{

//继续用y,提高y的利用率

y=p+1;

while((k[y]-k[y-1])==1)

{

for(q=-10;q<=10;q++)

{

if(get\_pixel\_color(i + 2, k[y]+q, page) == color)

{

y=-1;

break;

}

}

y--;

}

//满足右端尖角的第二个条件

if(y!=-1)

{

m--;//当作切线处理

}

}

}

else //是

{

m--;

}

}

judge = 1;//说明该边界是由多个点连城的一条线

}

if (judge == 0)//不是一条线，而是一个点！

{

//标记尖角型切线（只是做准备）

if ((k[p + 1] - k[p]) != 1 && (k[p] - k[p - 1]) != 1)

{

for (q = -5; q <= 5; q++)

{

if (get\_pixel\_color(i + 1, k[p] + q, page) == color)

{

special[0] = 1;

}

if (get\_pixel\_color(i - 1, k[p] + q, page) == color)

{

special[1] = 1;

}

}

if (special[0] == 1 && special[1] == 1)

{

;

}

else if(special[1] == 0)

{

m--;

t = i;

r = k[p];

}

//复原

special[1]=0;

special[0]=0;

}

m++;

//复原special数组

special[0] = 0;

special[1] = 0;

}

p++;

}

if (m % 2 == 0)

{

return 0;

}

else

{

return 1;

}

}

void ellipse\_(int centerx, int centery, int a, int b, int color, int page, int offsetx, int offsety)

//画椭圆

{

int x[5];

int y[5];

x[0] = centerx;

y[0] = centery + b;

x[1] = centerx - a;

y[1] = centery;

x[2] = centerx;

y[2] = centery - b;

x[3] = centerx + a;

y[3] = centery;

x[4] = centerx;

y[4] = centery + b;

Bezier(color, 5, page, x, y, 1);

fill(centerx + offsetx, centery + offsety, color, page, 1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

#include<graphics.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<dog.h>

#include<boy.h>

//上：k\*(y+350)-(y+353)\*(k-1)

//下：k\*(y+689)-(y+353)\*(k-1)

//左：k\*(x+442)-(x+524)\*(k-1)

//右：k\*(x+585)-(x+524)\*(k-1)

//原点为（524，353）

//画男生的头发

void boy\_hair(int x,int y,float k,int color,int page)

{

int x0[29];

int y0[29];

int x1[58];

int y1[58];

int x2[20];

int y2[20];

int i;

x=x-524;

y=y-353;

x0[0]=k\*(x+524);

y0[0]=k\*(y+353);

x0[1]=k\*(x+520);

y0[1]=k\*(y+357);

x0[2]=k\*(x+519);

y0[2]=k\*(y+360);

x0[3]=k\*(x+516);

y0[3]=k\*(y+363);

x0[4]=k\*(x+513);

y0[4]=k\*(y+367);

x0[5]=k\*(x+511);

y0[5]=k\*(y+369);

x0[6]=k\*(x+509);

y0[6]=k\*(y+372);

x0[7]=k\*(x+503);

y0[7]=k\*(y+375);

x0[8]=k\*(x+501);

y0[8]=k\*(y+377);

x0[9]=k\*(x+498);

y0[9]=k\*(y+379);

x0[10]=k\*(x+494);

y0[10]=k\*(y+380);

x0[11]=k\*(x+491);

y0[11]=k\*(y+381);

x0[12]=k\*(x+489);

y0[12]=k\*(y+381);

x0[13]=k\*(x+486);

y0[13]=k\*(y+381);

x0[14]=k\*(x+479);

y0[14]=k\*(y+381);

x0[15]=k\*(x+476);

y0[15]=k\*(y+381);

x0[16]=k\*(x+472);

y0[16]=k\*(y+381);

x0[17]=k\*(x+468);

y0[17]=k\*(y+380);

x0[18]=k\*(x+467);

y0[18]=k\*(y+379);

x0[19]=k\*(x+464);

y0[19]=k\*(y+379);

x0[20]=k\*(x+462);

y0[20]=k\*(y+378);

x0[21]=k\*(x+459);

y0[21]=k\*(y+378);

x0[22]=k\*(x+455);

y0[22]=k\*(y+377);

x0[23]=k\*(x+452);

y0[23]=k\*(y+377);

x0[24]=k\*(x+448);

y0[24]=k\*(y+376);

x0[25]=k\*(x+447);

y0[25]=k\*(y+373);

x0[26]=k\*(x+445);

y0[26]=k\*(y+370);

x0[27]=k\*(x+443);

y0[27]=k\*(y+367);

x0[28]=k\*(x+443);

y0[28]=k\*(y+363);

x1[0]=k\*(x+444);

y1[0]=k\*(y+360);

x1[1]=k\*(x+445);

y1[1]=k\*(y+357);

x1[2]=k\*(x+446);

y1[2]=k\*(y+353);

x1[3]=k\*(x+449);

y1[3]=k\*(y+349);

x1[4]=k\*(x+452);

y1[4]=k\*(y+346);

x1[5]=k\*(x+453);

y1[5]=k\*(y+343);

x1[6]=k\*(x+454);

y1[6]=k\*(y+341);

x1[7]=k\*(x+455);

y1[7]=k\*(y+338);

x1[8]=k\*(x+458);

y1[8]=k\*(y+335);

x1[9]=k\*(x+460);

y1[9]=k\*(y+332);

x1[10]=k\*(x+462);

y1[10]=k\*(y+329);

x1[11]=k\*(x+465);

y1[11]=k\*(y+323);

x1[12]=k\*(x+467);

y1[12]=k\*(y+321);

x1[13]=k\*(x+468);

y1[13]=k\*(y+321);

x1[14]=k\*(x+470);

y1[14]=k\*(y+321);

x1[15]=k\*(x+472);

x1[16]=k\*(x+475);

y1[16]=k\*(y+318);

x1[17]=k\*(x+479);

y1[17]=k\*(y+317);

x1[18]=k\*(x+481);

y1[18]=k\*(y+318);

x1[19]=k\*(x+486);

y1[19]=k\*(y+318);

x1[20]=k\*(x+490);

y1[20]=k\*(y+317);

x1[21]=k\*(x+494);

y1[21]=k\*(y+316);

x1[22]=k\*(x+500);

y1[22]=k\*(y+315);

x1[23]=k\*(x+504);

y1[23]=k\*(y+315);

x1[24]=k\*(x+507);

y1[24]=k\*(y+315);

x1[25]=k\*(x+510);

y1[25]=k\*(y+314);

x1[26]=k\*(x+512);

y1[26]=k\*(y+313);

x1[27]=k\*(x+515);

y1[27]=k\*(y+311);

x1[28]=k\*(x+517);

y1[28]=k\*(y+310);

x1[29]=k\*(x+519);

y1[29]=k\*(y+309);

x1[30]=k\*(x+521);

y1[30]=k\*(y+309);

x1[31]=k\*(x+523);

y1[31]=k\*(y+309);

x1[32]=k\*(x+525);

y1[32]=k\*(y+309);

x1[33]=k\*(x+525);

y1[33]=k\*(y+310);

x1[34]=k\*(x+525);

y1[34]=k\*(y+311);

x1[35]=k\*(x+525);

y1[35]=k\*(y+313);

x1[36]=k\*(x+525);

y1[36]=k\*(y+315);

x1[37]=k\*(x+530);

y1[37]=k\*(y+315);

x1[38]=k\*(x+532);

y1[38]=k\*(y+316);

x1[39]=k\*(x+536);

y1[39]=k\*(y+315);

x1[40]=k\*(x+542);

y1[40]=k\*(y+317);

x1[41]=k\*(x+549);

y1[41]=k\*(y+318);

x1[42]=k\*(x+554);

y1[42]=k\*(y+319);

x1[43]=k\*(x+558);

y1[43]=k\*(y+320);

x1[44]=k\*(x+562);

y1[44]=k\*(y+321);

x1[45]=k\*(x+567);

y1[45]=k\*(y+324);

x1[46]=k\*(x+569);

y1[46]=k\*(y+325);

x1[47]=k\*(x+572);

y1[47]=k\*(y+326);

x1[48]=k\*(x+576);

y1[48]=k\*(y+329);

x1[49]=k\*(x+582);

y1[49]=k\*(y+333);

x1[50]=k\*(x+586);

y1[50]=k\*(y+336);

x1[51]=k\*(x+590);

y1[51]=k\*(y+343);

x1[52]=k\*(x+590);

y1[52]=k\*(y+348);

x1[53]=k\*(x+593);

y1[53]=k\*(y+354);

x1[54]=k\*(x+595);

y1[54]=k\*(y+364);

x1[55]=k\*(x+596);

y1[55]=k\*(y+370);

x1[56]=k\*(x+597);

y1[56]=k\*(y+374);

x1[57]=k\*(x+596);

y1[57]=k\*(y+377);

x2[0]=k\*(x+593);

y2[0]=k\*(y+377);

x2[1]=k\*(x+591);

y2[1]=k\*(y+377);

x2[3]=k\*(x+589);

y2[3]=k\*(y+377);

x2[2]=k\*(x+588);

y2[2]=k\*(y+376);

x2[4]=k\*(x+581);

y2[4]=k\*(y+371);

x2[5]=k\*(x+578);

y2[5]=k\*(y+370);

x2[6]=k\*(x+574);

y2[6]=k\*(y+369);

x2[7]=k\*(x+565);

y2[7]=k\*(y+366);

x2[8]=k\*(x+561);

y2[8]=k\*(y+363);

x2[9]=k\*(x+554);

y2[9]=k\*(y+354);

x2[10]=k\*(x+549);

y2[10]=k\*(y+350);

x2[11]=k\*(x+546);

y2[11]=k\*(y+350);

x2[12]=k\*(x+544);

y2[12]=k\*(y+354);

x2[13]=k\*(x+542);

y2[13]=k\*(y+356);

x2[14]=k\*(x+541);

y2[14]=k\*(y+362);

x2[15]=k\*(x+539);

y2[15]=k\*(y+362);

x2[16]=k\*(x+534);

y2[16]=k\*(y+367);

x2[17]=k\*(x+530);

y2[17]=k\*(y+366);

x2[18]=k\*(x+527);

y2[18]=k\*(y+364);

x2[19]=k\*(x+524);

y2[19]=k\*(y+353);

//把初始顶点复原

for(i=0;i<29;i++)

{

x0[i]=x0[i]-(x+524)\*(k-1);

y0[i]=y0[i]-(y+353)\*(k-1);

}

for(i=0;i<58;i++)

{

x1[i]=x1[i]-(x+524)\*(k-1);

y1[i]=y1[i]-(y+353)\*(k-1);

}

for(i=0;i<20;i++)

{

x2[i]=x2[i]-(x+524)\*(k-1);

y2[i]=y2[i]-(y+353)\*(k-1);

}

Bezier(color,29,page,x0,y0,0);

Bezier(color,58,page,x1,y1,0);

Bezier(color,20,page,x2,y2,0);

}

//此处原点还是（524，353）

void boy\_face(int x,int y,float k,int color,int page)

{

int x0[25];

int y0[25];

int x1[44];

int y1[44];

int x2[19];

int y2[19];

int i=0;

x=x-524;

y=y-353;

//开始画左耳朵

x0[0]=k\*(x+444);

y0[0]=k\*(y+364);

x0[1]=k\*(x+438);

y0[1]=k\*(y+370);

x0[2]=k\*(x+437);

y0[2]=k\*(y+378);

x0[3]=k\*(x+437);

y0[3]=k\*(y+382);

x0[4]=k\*(x+436);

y0[4]=k\*(y+385);

x0[5]=k\*(x+436);

y0[5]=k\*(y+387);

x0[6]=k\*(x+436);

y0[6]=k\*(y+390);

x0[7]=k\*(x+434);

y0[7]=k\*(y+387);

x0[8]=k\*(x+433);

y0[8]=k\*(y+430);

x0[9]=k\*(x+428);

y0[9]=k\*(y+383);

x0[10]=k\*(x+425);

y0[10]=k\*(y+383);

x0[11]=k\*(x+422);

y0[11]=k\*(y+385);

x0[12]=k\*(x+421);

y0[12]=k\*(y+392);

x0[13]=k\*(x+419);

y0[13]=k\*(y+392);

x0[14]=k\*(x+419);

y0[14]=k\*(y+396);

x0[15]=k\*(x+419);

y0[15]=k\*(y+399);

x0[16]=k\*(x+420);

y0[16]=k\*(y+404);

x0[17]=k\*(x+423);

y0[17]=k\*(y+407);

x0[18]=k\*(x+425);

y0[18]=k\*(y+410);

x0[19]=k\*(x+427);

y0[19]=k\*(y+411);

x0[20]=k\*(x+432);

y0[20]=k\*(y+413);

x0[21]=k\*(x+437);

y0[21]=k\*(y+414);

x0[22]=k\*(x+440);

y0[22]=k\*(y+415);

x0[23]=k\*(x+444);

y0[23]=k\*(y+414);

x0[24]=k\*(x+446);

y0[24]=k\*(y+415);

//开始绘画脸下面的圆弧

x1[0]=k\*(x+446);

y1[0]=k\*(y+415);

x1[1]=k\*(x+448);

y1[1]=k\*(y+417);

x1[2]=k\*(x+450);

y1[2]=k\*(y+419);

x1[3]=k\*(x+451);

y1[3]=k\*(y+421);

x1[4]=k\*(x+452);

y1[4]=k\*(y+425);

x1[5]=k\*(x+453);

y1[5]=k\*(y+428);

x1[6]=k\*(x+454);

y1[6]=k\*(y+431);

x1[7]=k\*(x+455);

y1[7]=k\*(y+434);

x1[8]=k\*(x+457);

y1[8]=k\*(y+438);

x1[9]=k\*(x+458);

y1[9]=k\*(y+440);

x1[10]=k\*(x+461);

y1[10]=k\*(y+444);

x1[11]=k\*(x+465);

y1[11]=k\*(y+447);

x1[12]=k\*(x+467);

y1[12]=k\*(y+453);

x1[13]=k\*(x+470);

y1[13]=k\*(y+454);

x1[14]=k\*(x+474);

y1[14]=k\*(y+456);

x1[15]=k\*(x+479);

y1[15]=k\*(y+458);

x1[16]=k\*(x+485);

y1[16]=k\*(y+460);

x1[17]=k\*(x+489);

y1[17]=k\*(y+462);

x1[18]=k\*(x+494);

y1[18]=k\*(y+462);

x1[19]=k\*(x+497);

y1[19]=k\*(y+462);

x1[20]=k\*(x+501);

y1[20]=k\*(y+464);

x1[21]=k\*(x+505);

y1[21]=k\*(y+465);

x1[22]=k\*(x+510);

y1[22]=k\*(y+465);

x1[23]=k\*(x+516);

y1[23]=k\*(y+466);

x1[24]=k\*(x+520);

y1[24]=k\*(y+467);

x1[25]=k\*(x+523);

y1[25]=k\*(y+467);

x1[26]=k\*(x+526);

y1[26]=k\*(y+465);

x1[27]=k\*(x+531);

y1[27]=k\*(y+461);

x1[28]=k\*(x+536);

y1[28]=k\*(y+459);

x1[29]=k\*(x+540);

y1[29]=k\*(y+458);

x1[30]=k\*(x+543);

y1[30]=k\*(y+457);

x1[31]=k\*(x+547);

y1[31]=k\*(y+455);

x1[32]=k\*(x+553);

y1[32]=k\*(y+450);

x1[33]=k\*(x+556);

y1[33]=k\*(y+449);

x1[34]=k\*(x+560);

y1[34]=k\*(y+447);

x1[35]=k\*(x+564);

y1[35]=k\*(y+445);

x1[36]=k\*(x+571);

y1[36]=k\*(y+442);

x1[37]=k\*(x+574);

y1[37]=k\*(y+439);

x1[38]=k\*(x+577);

y1[38]=k\*(y+435);

x1[39]=k\*(x+583);

y1[39]=k\*(y+430);

x1[40]=k\*(x+583);

y1[40]=k\*(y+423);

x1[41]=k\*(x+584);

y1[41]=k\*(y+420);

x1[42]=k\*(x+584);

y1[42]=k\*(y+417);

x1[43]=k\*(x+585);

y1[43]=k\*(y+408);

//开始画右耳朵

x2[0]=k\*(x+585);

y2[0]=k\*(y+408);

x2[1]=k\*(x+587);

y2[1]=k\*(y+411);

x2[2]=k\*(x+590);

y2[2]=k\*(y+410);

x2[3]=k\*(x+593);

y2[3]=k\*(y+409);

x2[4]=k\*(x+597);

y2[4]=k\*(y+406);

x2[5]=k\*(x+599);

y2[5]=k\*(y+406);

x2[6]=k\*(x+602);

y2[6]=k\*(y+401);

x2[7]=k\*(x+604);

y2[7]=k\*(y+397);

x2[8]=k\*(x+605);

y2[8]=k\*(y+393);

x2[9]=k\*(x+605);

y2[9]=k\*(y+390);

x2[10]=k\*(x+604);

y2[10]=k\*(y+387);

x2[11]=k\*(x+603);

y2[11]=k\*(y+383);

x2[12]=k\*(x+601);

y2[12]=k\*(y+382);

x2[13]=k\*(x+598);

y2[13]=k\*(y+386);

x2[14]=k\*(x+596);

y2[14]=k\*(y+388);

x2[15]=k\*(x+594);

y2[15]=k\*(y+386);

x2[16]=k\*(x+595);

y2[16]=k\*(y+381);

x2[17]=k\*(x+595);

y2[17]=k\*(y+378);

x2[18]=k\*(x+596);

y2[18]=k\*(y+376);

//把放缩后的顶点复原

for(i=0;i<25;i++)

{

x0[i]=x0[i]-(x+524)\*(k-1);

y0[i]=y0[i]-(y+353)\*(k-1);

}

//把放缩后的顶点复原

for(i=0;i<44;i++)

{

x1[i]=x1[i]-(x+524)\*(k-1);

y1[i]=y1[i]-(y+353)\*(k-1);

}

//把放缩后的顶点复原

for(i=0;i<19;i++)

{

x2[i]=x2[i]-(x+524)\*(k-1);

y2[i]=y2[i]-(y+353)\*(k-1);

}

Bezier(color,25,page,x0,y0,0);

Bezier(color,44,page,x1,y1,0);

Bezier(color,19,page,x2,y2,0);

}

//原点与前面不发生改变（524，353）

void boy\_eye(int x,int y,float k,int color,int page)

{

int x0[28];

int y0[28];

int x1[31];

int y1[31];

int i=0;//用于后续循环

x=x-524;

y=y-353;

x0[0]=k\*(x+489);

y0[0]=k\*(y+392);

x0[1]=k\*(x+491);

y0[1]=k\*(y+392);

x0[2]=k\*(x+493);

y0[2]=k\*(y+392);

x0[3]=k\*(x+496);

y0[3]=k\*(y+391);

x0[4]=k\*(x+497);

y0[4]=k\*(y+393);

x0[5]=k\*(x+499);

y0[5]=k\*(y+394);

x0[6]=k\*(x+500);

y0[6]=k\*(y+395);

x0[7]=k\*(x+501);

y0[7]=k\*(y+397);

x0[8]=k\*(x+500);

y0[8]=k\*(y+401);

x0[9]=k\*(x+499);

y0[9]=k\*(y+403);

x0[10]=k\*(x+496);

y0[10]=k\*(y+405);

x0[11]=k\*(x+494);

y0[11]=k\*(y+406);

x0[12]=k\*(x+490);

y0[12]=k\*(y+407);

x0[13]=k\*(x+488);

y0[13]=k\*(y+406);

x0[14]=k\*(x+484);

y0[14]=k\*(y+406);

x0[15]=k\*(x+482);

y0[15]=k\*(y+407);

x0[16]=k\*(x+480);

y0[16]=k\*(y+405);

x0[17]=k\*(x+479);

y0[17]=k\*(y+404);

x0[18]=k\*(x+477);

y0[18]=k\*(y+402);

x0[19]=k\*(x+477);

y0[19]=k\*(y+400);

x0[20]=k\*(x+476);

y0[20]=k\*(y+398);

x0[21]=k\*(x+477);

y0[21]=k\*(y+395);

x0[22]=k\*(x+478);

y0[22]=k\*(y+394);

x0[23]=k\*(x+479);

y0[23]=k\*(y+393);

x0[24]=k\*(x+482);

y0[24]=k\*(y+393);

x0[25]=k\*(x+485);

y0[25]=k\*(y+392);

x0[26]=k\*(x+486);

y0[26]=k\*(y+392);

x0[27]=k\*(x+489);

y0[27]=k\*(y+392);

//开始画右面的眼睛

x1[0]=k\*(x+544);

y1[0]=k\*(y+393);

x1[1]=k\*(x+542);

y1[1]=k\*(y+393);

for(i=0;i<5;i++)

{

x1[2+i]=k\*(x+543+i);

y1[2+i]=k\*(y+393);

}

x1[7]=k\*(x+548);

y1[7]=k\*(y+394);

x1[8]=k\*(x+550);

y1[8]=k\*(y+394);

x1[9]=k\*(x+553);

y1[9]=k\*(y+394);

x1[10]=k\*(x+555);

y1[10]=k\*(y+396);

x1[11]=k\*(x+556);

y1[11]=k\*(y+398);

x1[12]=k\*(x+555);

y1[12]=k\*(y+401);

x1[13]=k\*(x+551);

y1[13]=k\*(y+405);

x1[14]=k\*(x+547);

y1[14]=k\*(y+406);

x1[15]=k\*(x+545);

y1[15]=k\*(y+406);

x1[16]=k\*(x+542);

y1[16]=k\*(y+407);

for(i=0;i<7;i++)

{

x1[17+i]=k\*(x+541-i);

y1[17+i]=k\*(y+407);

}

x1[24]=k\*(x+533);

y1[24]=k\*(y+402);

x1[25]=k\*(x+532);

y1[25]=k\*(y+400);

x1[26]=k\*(x+534);

y1[26]=k\*(y+398);

x1[27]=k\*(x+536);

y1[27]=k\*(y+393);

x1[28]=k\*(x+538);

y1[28]=k\*(y+392);

x1[29]=k\*(x+541);

y1[29]=k\*(y+392);

x1[30]=k\*(x+544);

y1[30]=k\*(y+393);

//把放缩后的顶点复原

for(i=0;i<28;i++)

{

x0[i]=x0[i]-(x+524)\*(k-1);

y0[i]=y0[i]-(y+353)\*(k-1);

}

//把放缩后的顶点复原

for(i=0;i<31;i++)

{

x1[i]=x1[i]-(x+524)\*(k-1);

y1[i]=y1[i]-(y+353)\*(k-1);

}

Bezier(color,28,page,x0,y0,0);

Bezier(color,31,page,x1,y1,0);

}

void boy\_body(int x,int y,float k,int color,int page)

{

int x0[28];

int y0[28];

int x1[31];

int y1[31];

int x2[15];

int y2[15];

int i=0;//用于后续循环

x=x-524;

y=y-353;

x0[0]=k\*(x+482);

y0[0]=k\*(y+488);

x0[1]=k\*(x+481);

y0[1]=k\*(y+491);

x0[1]=k\*(x+479);

y0[1]=k\*(y+496);

x0[2]=k\*(x+477);

y0[2]=k\*(y+497);

x0[3]=k\*(x+475);

y0[3]=k\*(y+502);

x0[4]=k\*(x+472);

y0[4]=k\*(y+510);

x0[5]=k\*(x+471);

y0[5]=k\*(y+519);

x0[6]=k\*(x+468);

y0[6]=k\*(y+528);

x0[7]=k\*(x+467);

y0[7]=k\*(y+538);

x0[8]=k\*(x+465);

y0[8]=k\*(y+543);

x0[9]=k\*(x+464);

y0[9]=k\*(y+551);

x0[10]=k\*(x+463);

y0[10]=k\*(y+557);

x0[11]=k\*(x+463);

y0[11]=k\*(y+569);

x0[12]=k\*(x+463);

y0[12]=k\*(y+579);

//把放缩后的顶点复原

for(i=0;i<13;i++)

{

x0[i]=x0[i]-(x+524)\*(k-1);

y0[i]=y0[i]-(y+353)\*(k-1);

}

Bezier(color,13,page,x0,y0,0);

x1[0]=k\*(x+542);

y1[0]=k\*(y+484);

x1[1]=k\*(x+547);

y1[1]=k\*(y+495);

x1[2]=k\*(x+552);

y1[2]=k\*(y+501);

x1[3]=k\*(x+555);

y1[3]=k\*(y+509);

x1[4]=k\*(x+558);

y1[4]=k\*(y+516);

x1[5]=k\*(x+560);

y1[5]=k\*(y+519);

x1[6]=k\*(x+563);

y1[6]=k\*(y+527);

x1[7]=k\*(x+565);

y1[7]=k\*(y+537);

x1[8]=k\*(x+567);

y1[8]=k\*(y+548);

x1[9]=k\*(x+572);

y1[9]=k\*(y+559);

x1[10]=k\*(x+575);

y1[10]=k\*(y+569);

x1[11]=k\*(x+571);

y1[11]=k\*(y+578);

//把放缩后的顶点复原

for(i=0;i<12;i++)

{

x1[i]=x1[i]-(x+524)\*(k-1);

y1[i]=y1[i]-(y+353)\*(k-1);

}

Bezier(color,12,page,x1,y1,0);

x2[0]=k\*(x+463);

y2[0]=k\*(y+579);

x2[1]=k\*(x+465);

y2[1]=k\*(y+582);

x2[2]=k\*(x+472);

y2[2]=k\*(y+585);

x2[3]=k\*(x+479);

y2[3]=k\*(y+587);

x2[4]=k\*(x+481);

y2[4]=k\*(y+587);

x2[5]=k\*(x+486);

y2[5]=k\*(y+591);

x2[6]=k\*(x+495);

y2[6]=k\*(y+590);

x2[7]=k\*(x+501);

y2[7]=k\*(y+592);

x2[8]=k\*(x+510);

y2[8]=k\*(y+592);

x2[9]=k\*(x+522);

y2[9]=k\*(y+591);

x2[10]=k\*(x+532);

y2[10]=k\*(y+590);

x2[11]=k\*(x+541);

y2[11]=k\*(y+588);

x2[12]=k\*(x+549);

y2[12]=k\*(y+585);

x2[13]=k\*(x+556);

y2[13]=k\*(y+581);

x2[14]=k\*(x+571);

y2[14]=k\*(y+578);

//把放缩后的顶点复原

for(i=0;i<12;i++)

{

x2[i]=x2[i]-(x+524)\*(k-1);

y2[i]=y2[i]-(y+353)\*(k-1);

}

Bezier(color,15,page,x2,y2,0);

}

void boy\_leg(int x,int y,float k,int color,int page)

{

int i=0;

x=x-524;

y=y-353;

//左侧短裤

for(i=-2;i<=2;i++)

{

line\_drawing(k\*(x+482)-(x+524)\*(k-1),k\*(y+589)-(y+353)\*(k-1),k\*(x+476)-(x+524)\*(k-1),k\*(y+623)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+499)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),k\*(x+502)-(x+524)\*(k-1),k\*(y+593)-(y+353)\*(k-1),color,page,1);

}

for(i=-3;i<3;i++)

{

line\_drawing(k\*(x+476)-(x+524)\*(k-1),k\*(y+623)-(y+353)\*(k-1),k\*(x+499)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),color,page,1);

}

//右侧短裤

for(i=-2;i<=2;i++)

{

line\_drawing(k\*(x+531)-(x+521)\*(k-1),k\*(y+591)-(y+353)\*(k-1),k\*(x+533)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+556)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),k\*(x+548)-(x+524)\*(k-1),k\*(y+592)-(y+348)\*(k-1),color,page,1);

}

for(i=-3;i<3;i++)

{

line\_drawing(k\*(x+533)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),k\*(x+556)-(x+524)\*(k-1),k\*(y+622)-(y+353)\*(k-1),color,page,1);

}

//左腿

for(i=-2;i<=2;i++)

{

line\_drawing(k\*(x+481)-(x+524)\*(k-1),k\*(y+626)-(y+353)\*(k-1),k\*(x+472)-(x+524)\*(k-1),k\*(y+668)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+492)-(x+524)\*(k-1),k\*(y+624)-(y+353)\*(k-1),k\*(x+487)-(x+524)\*(k-1),k\*(y+668)-(y+353)\*(k-1),color,page,1);

}

//右腿

for(i=-2;i<=2;i++)

{

line\_drawing(k\*(x+541)-(x+524)\*(k-1),k\*(y+624)-(y+353)\*(k-1),k\*(x+543)-(x+524)\*(k-1),k\*(y+668)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+552)-(x+524)\*(k-1),k\*(y+624)-(y+353)\*(k-1),k\*(x+554)-(x+524)\*(k-1),k\*(y+668)-(y+353)\*(k-1),color,page,1);

}

//左脚

for(i=-3;i<=3;i++)

{

line\_drawing(k\*(x+487)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),k\*(x+461)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+461)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),k\*(x+460)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+460)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),k\*(x+490)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+490)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),k\*(x+487)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),color,page,1);

}

//右脚

for(i=-3;i<=3;i++)

{

line\_drawing(k\*(x+538)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),k\*(x+537)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+537)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),k\*(x+566)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+566)-(x+524)\*(k-1),k\*(y+682)-(y+353)\*(k-1),k\*(x+568)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+568)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),k\*(x+538)-(x+524)\*(k-1),k\*(y+669)-(y+353)\*(k-1),color,page,1);

}

}

void boy\_arm(int x,int y,float k,int color,int page)

{

int i=0;

int x0[10];

int y0[10];

int x1[9];

int y1[9];

int x2[10];

int y2[10];

x=x-524;

y=y-353;

//左肩

x0[0]=k\*(x+480);

y0[0]=k\*(y+482);

x0[1]=k\*(x+478);

y0[1]=k\*(y+483);

x0[2]=k\*(x+476);

y0[2]=k\*(y+485);

x0[3]=k\*(x+474);

y0[3]=k\*(y+487);

x0[4]=k\*(x+472);

y0[4]=k\*(y+489);

x0[5]=k\*(x+471);

y0[5]=k\*(y+492);

x0[6]=k\*(x+468);

y0[6]=k\*(y+494);

x0[7]=k\*(x+464);

y0[7]=k\*(y+497);

x0[8]=k\*(x+462);

y0[8]=k\*(y+499);

x0[9]=k\*(x+460);

y0[9]=k\*(y+502);

line\_drawing(k\*(x+460)-(x+524)\*(k-1),k\*(y+502)-(y+353)\*(k-1),k\*(x+476)-(x+524)\*(k-1),k\*(y+502)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+460)-(x+524)\*(k-1),k\*(y+503)-(y+353)\*(k-1),k\*(x+476)-(x+524)\*(k-1),k\*(y+503)-(y+353)\*(k-1),color,page,1);

//line\_drawing(k\*(x+460)-(x+524)\*(k-1),k\*(y+504)-(y+353)\*(k-1),k\*(x+480)-(x+524)\*(k-1),k\*(y+504)-(y+353)\*(k-1),color,page,1);

//右肩

x1[0]=k\*(x+551);

y1[0]=k\*(y+482);

x1[1]=k\*(x+554);

y1[1]=k\*(y+483);

x1[2]=k\*(x+555);

y1[2]=k\*(y+484);

x1[3]=k\*(x+557);

y1[3]=k\*(y+486);

x1[4]=k\*(x+559);

y1[4]=k\*(y+488);

x1[5]=k\*(x+563);

y1[5]=k\*(y+492);

x1[6]=k\*(x+566);

y1[6]=k\*(y+497);

x1[7]=k\*(x+568);

y1[7]=k\*(y+501);

x1[8]=k\*(x+570);

y1[8]=k\*(y+502);

line\_drawing(k\*(x+570)-(x+524)\*(k-1),k\*(y+502)-(y+353)\*(k-1),k\*(x+554)-(x+524)\*(k-1),k\*(y+502)-(y+353)\*(k-1),color,page,1);

line\_drawing(k\*(x+570)-(x+524)\*(k-1),k\*(y+503)-(y+353)\*(k-1),k\*(x+554)-(x+524)\*(k-1),k\*(y+503)-(y+353)\*(k-1),color,page,1);

//line\_drawing(k\*(x+570)-(x+524)\*(k-1),k\*(y+504),k\*(x+554)-(x+524)\*(k-1),k\*(y+504),color,page,1);

//把放缩后的顶点复原

for(i=0;i<10;i++)

{

x0[i]=x0[i]-(x+524)\*(k-1);

y0[i]=y0[i]-(y+353)\*(k-1);

}

Bezier(color,10,page,x0,y0,0);

for(i=0;i<9;i++)

{

x1[i]=x1[i]-(x+524)\*(k-1);

y1[i]=y1[i]-(y+353)\*(k-1);

}

Bezier(color,9,page,x1,y1,0);

//左臂

line\_drawing(k\*(x+464)-(x+524)\*(k-1),k\*(y+505)-(y+502)\*(k-1),k\*(x+447)-(x+524)\*(k-1),k\*(y+579)-(y+502)\*(k-1),color,page,1);

line\_drawing(k\*(x+447)-(x+524)\*(k-1),k\*(y+579)-(y+502)\*(k-1),k\*(x+454)-(x+524)\*(k-1),k\*(y+591)-(y+502)\*(k-1),color,page,1);

line\_drawing(k\*(x+454)-(x+524)\*(k-1),k\*(y+591)-(y+502)\*(k-1),k\*(x+460)-(y+524)\*(k-1),k\*(y+568)-(y+502)\*(k-1),color,page,1);

line\_drawing(k\*(x+564)-(x+524)\*(k-1),k\*(y+505)-(y+502)\*(k-1),k\*(x+590)-(x+524)\*(k-1),k\*(y+579)-(y+502)\*(k-1),color,page,1);

line\_drawing(k\*(x+590)-(x+524)\*(k-1),k\*(y+579)-(y+502)\*(k-1),k\*(x+578)-(x+524)\*(k-1),k\*(y+592)-(y+502)\*(k-1),color,page,1);

line\_drawing(k\*(x+578)-(x+524)\*(k-1),k\*(y+592)-(y+502)\*(k-1),k\*(x+570)-(x+524)\*(k-1),k\*(y+562)-(y+502)\*(k-1),color,page,1);

//mouth

x2[0]=k\*(x+504);

y2[0]=k\*(y+448);

x2[1]=k\*(x+509);

y2[1]=k\*(y+453);

x2[2]=k\*(x+512);

y2[2]=k\*(y+456);

x2[3]=k\*(x+516);

y2[3]=k\*(y+456);

x2[4]=k\*(x+517);

y2[4]=k\*(y+456);

x2[5]=k\*(x+518);

y2[5]=k\*(y+457);

x2[6]=k\*(x+512);

y2[6]=k\*(y+456);

x2[7]=k\*(x+522);

y2[7]=k\*(y+456);

x2[8]=k\*(x+526);

y2[8]=k\*(y+453);

x2[9]=k\*(x+530);

y2[9]=k\*(y+451);

//把放缩后的顶点复原

for(i=0;i<10;i++)

{

x2[i]=x2[i]-(x+524)\*(k-1);

y2[i]=y2[i]-(y+353)\*(k-1);

}

Bezier(color,10,page,x2,y2,0);

}

void boy(int x,int y,float k,int color,int page)

{

boy\_face(x,y,k,RED\_,page);

boy\_hair(x,y,k,RED\_,page);

boy\_eye(x,y,k,RED\_,page);

boy\_body(x,y-20,k,RED\_,page);

boy\_leg(x,y-20,k,RED\_,page);

boy\_arm(x,y-20,k,RED\_,page);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

#include<graphics.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<dog.h>

#include<boy.h>

//上：362

//下：662

//左：207

//右：683

//原点在（304，618）

void car(int x,int y,float k,int color,int page)

{

int x0[29];

int y0[29];

int x1[16];

int y1[16];

int x2[12];

int y2[12];

int x3[26];

int y3[26];

int x4[21];

int y4[21];

int i;

x=x-304;

y=y-618;

line\_drawing(k\*(x+304)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),k\*(x+236)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),color,page,1);

x0[0]=k\*(x+236);

y0[0]=k\*(y+618);

x0[1]=k\*(x+228);

y0[1]=k\*(y+614);

x0[2]=k\*(x+223);

y0[2]=k\*(y+606);

x0[3]=k\*(x+222);

y0[3]=k\*(y+596);

x0[4]=k\*(x+225);

y0[4]=k\*(y+589);

x0[5]=k\*(x+226);

y0[5]=k\*(y+585);

x0[6]=k\*(x+228);

y0[6]=k\*(y+583);

x0[7]=k\*(x+229);

y0[7]=k\*(y+580);

x0[8]=k\*(x+231);

y0[8]=k\*(y+573);

x0[9]=k\*(x+236);

y0[9]=k\*(y+563);

x0[10]=k\*(x+239);

y0[10]=k\*(y+555);

x0[11]=k\*(x+242);

y0[11]=k\*(y+549);

x0[12]=k\*(x+245);

y0[12]=k\*(y+543);

x0[13]=k\*(x+251);

y0[13]=k\*(y+533);

x0[14]=k\*(x+256);

y0[14]=k\*(y+527);

x0[15]=k\*(x+261);

y0[15]=k\*(y+519);

x0[16]=k\*(x+265);

y0[16]=k\*(y+513);

x0[17]=k\*(x+274);

y0[17]=k\*(y+502);

x0[18]=k\*(x+282);

y0[18]=k\*(y+497);

x0[19]=k\*(x+287);

y0[19]=k\*(y+493);

x0[20]=k\*(x+292);

y0[20]=k\*(y+490);

x0[21]=k\*(x+296);

y0[21]=k\*(y+486);

//把放缩后的顶点复原

for(i=0;i<22;i++)

{

x0[i]=x0[i]-(x+304)\*(k-1);

y0[i]=y0[i]-(y+618)\*(k-1);

}

Bezier(color,22,page,x0,y0,0);

line\_drawing(k\*(x+296)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),k\*(x+314)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+314)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),k\*(x+314)-(x+304)\*(k-1),k\*(y+460)-(y+618)\*(k-1),color,page,1);

x1[0]=k\*(x+314);

y1[0]=k\*(y+460);

x1[1]=k\*(x+317);

y1[1]=k\*(y+454);

x1[2]=k\*(x+320);

y1[2]=k\*(y+442);

x1[3]=k\*(x+324);

y1[3]=k\*(y+434);

x1[4]=k\*(x+326);

y1[4]=k\*(y+426);

x1[5]=k\*(x+330);

y1[5]=k\*(y+419);

x1[6]=k\*(x+334);

y1[6]=k\*(y+411);

x1[7]=k\*(x+339);

y1[7]=k\*(y+405);

x1[8]=k\*(x+343);

y1[8]=k\*(y+400);

x1[9]=k\*(x+349);

y1[9]=k\*(y+399);

x1[10]=k\*(x+351);

y1[10]=k\*(y+394);

x1[11]=k\*(x+356);

y1[11]=k\*(y+391);

x1[12]=k\*(x+364);

y1[12]=k\*(y+386);

x1[13]=k\*(x+366);

y1[13]=k\*(y+384);

x1[14]=k\*(x+374);

y1[14]=k\*(y+383);

x1[15]=k\*(x+377);

y1[15]=k\*(y+382);

//把放缩后的顶点复原

for(i=0;i<16;i++)

{

x1[i]=x1[i]-(x+304)\*(k-1);

y1[i]=y1[i]-(y+618)\*(k-1);

}

//Bezier(color,16,page,x1,y1,0);

//line\_drawing(k\*(x+296)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),k\*(x+314)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),color,page,1);

//line\_drawing(k\*(x+314)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),k\*(x+314)-(x+304)\*(k-1),k\*(y+460)-(y+618)\*(k-1),color,page,1);

x2[0]=k\*(x+314);

y2[0]=k\*(y+460);

x2[1]=k\*(x+315);

y2[1]=k\*(y+452);

x2[2]=k\*(x+317);

y2[2]=k\*(y+439);

x2[3]=k\*(x+319);

y2[3]=k\*(y+432);

x2[4]=k\*(x+323);

y2[4]=k\*(y+428);

x2[5]=k\*(x+328);

y2[5]=k\*(y+422);

x2[6]=k\*(x+336);

y2[6]=k\*(y+411);

x2[7]=k\*(x+341);

y2[7]=k\*(y+402);

x2[8]=k\*(x+348);

y2[8]=k\*(y+399);

x2[9]=k\*(x+355);

y2[9]=k\*(y+390);

x2[10]=k\*(x+365);

y2[10]=k\*(y+387);

x2[11]=k\*(x+366);

y2[11]=k\*(y+384);

//把放缩后的顶点复原

for(i=0;i<12;i++)

{

x2[i]=x2[i]-(x+304)\*(k-1);

y2[i]=y2[i]-(y+618)\*(k-1);

}

Bezier(color,12,page,x2,y2,0);

line\_drawing(k\*(x+366)-(x+304)\*(k-1),k\*(y+384)-(y+618)\*(k-1),k\*(x+478)-(x+304)\*(k-1),k\*(y+384)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+439)-(x+304)\*(k-1),k\*(y+384)-(y+618)\*(k-1),k\*(x+460)-(x+304)\*(k-1),k\*(y+311)-(y+618)\*(k-1),color,page,1);

x3[0]=k\*(x+478);

y3[0]=k\*(y+384);

x3[1]=k\*(x+486);

y3[1]=k\*(y+384);

x3[2]=k\*(x+492);

y3[2]=k\*(y+384);

x3[3]=k\*(x+496);

y3[3]=k\*(y+384);

x3[4]=k\*(x+502);

y3[4]=k\*(y+386);

x3[5]=k\*(x+511);

y3[5]=k\*(y+388);

x3[6]=k\*(x+516);

y3[6]=k\*(y+390);

x3[7]=k\*(x+520);

y3[7]=k\*(y+392);

x3[8]=k\*(x+531);

y3[8]=k\*(y+397);

x3[9]=k\*(x+538);

y3[9]=k\*(y+402);

x3[10]=k\*(x+543);

y3[10]=k\*(y+408);

x3[11]=k\*(x+547);

y3[11]=k\*(y+412);

x3[12]=k\*(x+554);

y3[12]=k\*(y+415);

x3[13]=k\*(x+558);

y3[13]=k\*(y+418);

x3[14]=k\*(x+556);

y3[14]=k\*(y+419);

x3[15]=k\*(x+560);

y3[15]=k\*(y+420);

x3[16]=k\*(x+564);

y3[16]=k\*(y+424);

x3[17]=k\*(x+566);

y3[17]=k\*(y+427);

x3[18]=k\*(x+569);

y3[18]=k\*(y+432);

x3[19]=k\*(x+574);

y3[19]=k\*(y+437);

x3[20]=k\*(x+578);

y3[20]=k\*(y+441);

x3[21]=k\*(x+580);

y3[21]=k\*(y+446);

x3[22]=k\*(x+584);

y3[22]=k\*(y+452);

x3[23]=k\*(x+589);

y3[23]=k\*(y+457);

x3[24]=k\*(x+588);

y3[24]=k\*(y+462);

x3[25]=k\*(x+589);

y3[25]=k\*(y+460);

//把放缩后的顶点复原

for(i=0;i<26;i++)

{

x3[i]=x3[i]-(x+304)\*(k-1);

y3[i]=y3[i]-(y+618)\*(k-1);

}

Bezier(color,26,page,x3,y3,0);

line\_drawing(k\*(x+589)-(x+304)\*(k-1),k\*(y+460)-(y+618)\*(k-1),k\*(x+589)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+589)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),k\*(x+619)-(x+304)\*(k-1),k\*(y+486)-(y+618)\*(k-1),color,page,1);

x4[0]=k\*(x+619);

y4[0]=k\*(y+486);

x4[1]=k\*(x+629);

y4[1]=k\*(y+490);

x4[2]=k\*(x+633);

y4[2]=k\*(y+492);

x4[3]=k\*(x+635);

y4[3]=k\*(y+495);

x4[4]=k\*(x+641);

y4[4]=k\*(y+502);

x4[5]=k\*(x+643);

y4[5]=k\*(y+507);

x4[6]=k\*(x+648);

y4[6]=k\*(y+511);

x4[7]=k\*(x+654);

y4[7]=k\*(y+519);

x4[8]=k\*(x+656);

y4[8]=k\*(y+523);

x4[9]=k\*(x+658);

y4[9]=k\*(y+528);

x4[10]=k\*(x+662);

y4[10]=k\*(y+534);

x4[11]=k\*(x+664);

y4[11]=k\*(y+543);

x4[12]=k\*(x+665);

y4[12]=k\*(y+551);

x4[13]=k\*(x+664);

y4[13]=k\*(y+560);

x4[14]=k\*(x+663);

y4[14]=k\*(y+658);

x4[15]=k\*(x+664);

y4[15]=k\*(y+577);

x4[16]=k\*(x+660);

y4[16]=k\*(y+585);

x4[17]=k\*(x+659);

y4[17]=k\*(y+592);

x4[18]=k\*(x+657);

y4[18]=k\*(y+596);

x4[19]=k\*(x+654);

y4[19]=k\*(y+603);

x4[20]=k\*(x+639);

y4[20]=k\*(y+618);

//把放缩后的顶点复原

for(i=0;i<21;i++)

{

x4[i]=x4[i]-(x+304)\*(k-1);

y4[i]=y4[i]-(y+618)\*(k-1);

}

Bezier(color,21,page,x4,y4,0);

line\_drawing(k\*(x+639)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),k\*(x+587)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),color,page,1);

circle\_without\_drawing(k\*(x+547)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),40,color,page);

circle\_without\_drawing(k\*(x+344)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),40,color,page);

circle\_without\_drawing(k\*(x+547)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),20,color,page);

circle\_without\_drawing(k\*(x+344)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),20,color,page);

line\_drawing(k\*(x+384)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),k\*(x+507)-(x+304)\*(k-1),k\*(y+618)-(y+618)\*(k-1),color,page,1);

}

void car\_light(int x,int y,float k,int color,int page)

{

int i=0;

int x0[19];

int y0[19];

x=x-304;

y=y-618;

x0[0]=k\*(x+232);

y0[0]=k\*(y+567);

x0[1]=k\*(x+226);

y0[1]=k\*(y+563);

x0[2]=k\*(x+220);

y0[2]=k\*(y+556);

x0[3]=k\*(x+215);

y0[3]=k\*(y+552);

x0[4]=k\*(x+211);

y0[4]=k\*(y+547);

x0[5]=k\*(x+209);

y0[5]=k\*(y+540);

x0[6]=k\*(x+208);

y0[6]=k\*(y+532);

x0[7]=k\*(x+209);

y0[7]=k\*(y+524);

x0[8]=k\*(x+213);

y0[8]=k\*(y+516);

x0[9]=k\*(x+219);

y0[9]=k\*(y+508);

x0[10]=k\*(x+227);

y0[10]=k\*(y+504);

x0[11]=k\*(x+235);

y0[11]=k\*(y+504);

x0[12]=k\*(x+241);

y0[12]=k\*(y+503);

x0[13]=k\*(x+246);

y0[13]=k\*(y+501);

x0[14]=k\*(x+250);

y0[14]=k\*(y+503);

x0[15]=k\*(x+257);

y0[15]=k\*(y+507);

x0[16]=k\*(x+259);

y0[16]=k\*(y+508);

x0[17]=k\*(x+264);

y0[17]=k\*(y+508);

x0[18]=k\*(x+266);

y0[18]=k\*(y+512);

//把放缩后的顶点复原

for(i=0;i<19;i++)

{

x0[i]=x0[i]-(x+304)\*(k-1);

y0[i]=y0[i]-(y+618)\*(k-1);

}

Bezier(color,19,page,x0,y0,0);

}

void car\_windows(int x,int y,float k,int color,int page)

{

int i=0;

int x0[14];

int y0[14];

int x1[14];

int y1[14];

x=x-304;

y=y-618;

x0[0]=k\*(x+340);

y0[0]=k\*(y+495);

x0[1]=k\*(x+344);

y0[1]=k\*(y+482);

x0[2]=k\*(x+348);

y0[2]=k\*(y+472);

x0[3]=k\*(x+352);

y0[3]=k\*(y+462);

x0[4]=k\*(x+357);

y0[4]=k\*(y+454);

x0[5]=k\*(x+359);

y0[5]=k\*(y+448);

x0[6]=k\*(x+363);

y0[6]=k\*(y+442);

x0[7]=k\*(x+371);

y0[7]=k\*(y+438);

x0[8]=k\*(x+376);

y0[8]=k\*(y+433);

x0[9]=k\*(x+380);

y0[9]=k\*(y+431);

x0[10]=k\*(x+389);

y0[10]=k\*(y+427);

x0[11]=k\*(x+395);

y0[11]=k\*(y+426);

x0[12]=k\*(x+401);

y0[12]=k\*(y+424);

x0[13]=k\*(x+409);

y0[13]=k\*(y+422);

//把放缩后的顶点复原

for(i=0;i<14;i++)

{

x0[i]=x0[i]-(x+304)\*(k-1);

y0[i]=y0[i]-(y+618)\*(k-1);

}

Bezier(color,14,page,x0,y0,0);

x1[0]=k\*(x+463);

y1[0]=k\*(y+422);

x1[1]=k\*(x+473);

y1[1]=k\*(y+422);

x1[2]=k\*(x+481);

y1[2]=k\*(y+423);

x1[3]=k\*(x+487);

y1[3]=k\*(y+423);

x1[4]=k\*(x+492);

y1[4]=k\*(y+424);

x1[5]=k\*(x+497);

y1[5]=k\*(y+427);

x1[6]=k\*(x+503);

y1[6]=k\*(y+431);

x1[7]=k\*(x+508);

y1[7]=k\*(y+434);

x1[8]=k\*(x+510);

y1[8]=k\*(y+436);

x1[9]=k\*(x+516);

y1[9]=k\*(y+439);

x1[10]=k\*(x+520);

y1[10]=k\*(y+444);

x1[11]=k\*(x+524);

y1[11]=k\*(y+448);

x1[12]=k\*(x+526);

y1[12]=k\*(y+489);

x1[13]=k\*(x+533);

y1[13]=k\*(y+495);

//把放缩后的顶点复原

for(i=0;i<14;i++)

{

x1[i]=x1[i]-(x+304)\*(k-1);

y1[i]=y1[i]-(y+618)\*(k-1);

}

Bezier(color,14,page,x1,y1,0);

line\_drawing(k\*(x+340)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),k\*(x+409)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+409)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),k\*(x+409)-(x+304)\*(k-1),k\*(y+422)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+463)-(x+304)\*(k-1),k\*(y+422)-(y+618)\*(k-1),k\*(x+463)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),color,page,1);

line\_drawing(k\*(x+463)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),k\*(x+533)-(x+304)\*(k-1),k\*(y+495)-(y+618)\*(k-1),color,page,1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：detect

\*\*文件目的：检测有没有其他类型的字符并改变，记录数字

\*\*时间复杂度：正比于字数和特殊字符数

\*\*空间复杂度：正比于字数和特殊字符数

\*/

#include<stdio.h>

#include<string.h>

#include<detect.h>

//还需添加空格，标点等特殊字符的处理，空格是去掉或换为句号，标点是两倍

int detect(char\* input,int\* strategy)

{

int sum=0;

\*strategy=CHINESE;

int length=strlen(input);

int i,j;

for(i=0;i<length;i++)

{

if(input[i]<0)

{

continue;

}

else if(input[i]>='0'&&input[i]<='9')

{

sum++;

\*strategy=CALCULATE;

//将阿拉伯数字重复一遍，以适应GB2312的字节数

//strcpy(input,"啊哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈");//test

/\*

input[length+1]='\0';

for(j=length-1;j>=i+1;j--)

{

input[j+1]=input[j];

}

input[i+1]=input[i];

length++;

\*/

}

else if(input[i]=='+'||input[i]=='='||input[i]=='-'||input[i]=='\*'||input[i]=='/')

{

sum++;

\*strategy=CALCULATE;

}

else if((input[i]>='a'&&input[i]<='z')||(input[i]>='A'&&input[i]<='Z'))

{

//strcpy(input,"这是英语！");//test

sum++;

//\*strategy=ENGLISH;

//将英文字母重复一遍，以适应GB2312的字节数

/\*

input[length+1]=0;

for(j=length-1;j>=i+1;j--)

{

input[j+1]=input[j];

}

input[i+1]=input[i];

length++;

\*/

}

}

return sum;//返回ASC2字符的总数

}

int ddetect(char\* input)

{

int length=strlen(input);

int i,j;

for(i=0;i<length;i++)

{

if(input[i]<0)

{

continue;

}

else

{

input[length+1]='\0';

for(j=length-1;j>=i+1;j--)

{

input[j+1]=input[j];

}

input[i+1]=input[i];

length++;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

#include<graphics.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<dog.h>

//上：70

//下：470

//左：216

//右：610

//画图原点位于（414，245）

void dog\_body(int x,int y,float k,int color,int page)

{

int x0[432];

int y0[432];

int i;

//严格记录图形所在区域

int posi[4]={0};

posi[0]=70;//上

posi[1]=470;//下

posi[2]=216;//左

posi[3]=610;//右

x=x-414;

y=y-245;

x0[0]=k\*(x+414);

y0[0]=k\*(y+245);

x0[1]=k\*(x+406);

y0[1]=k\*(y+252);

x0[2]=k\*(x+402);

y0[2]=k\*(y+255);

x0[3]=k\*(x+398);

y0[3]=k\*(y+256);

x0[4]=k\*(x+394);

y0[4]=k\*(y+257);

x0[5]=k\*(x+386);

y0[5]=k\*(y+259);

x0[6]=k\*(x+380);

y0[6]=k\*(y+261);

x0[7]=k\*(x+362);

y0[7]=k\*(y+262);

x0[8]=k\*(x+337);

y0[8]=k\*(y+262);

x0[9]=k\*(x+326);

y0[9]=k\*(y+261);

x0[10]=k\*(x+320);

y0[10]=k\*(y+259);

x0[11]=k\*(x+314);

y0[11]=k\*(y+258);

x0[12]=k\*(x+308);

y0[12]=k\*(y+258);

x0[13]=k\*(x+299);

y0[13]=k\*(y+258);

x0[14]=k\*(x+287);

y0[14]=k\*(y+258);

x0[15]=k\*(x+279);

y0[15]=k\*(y+258);

x0[16]=k\*(x+274);

y0[16]=k\*(y+256);

x0[17]=k\*(x+266);

y0[17]=k\*(y+243);

x0[18]=k\*(x+264);

y0[18]=k\*(y+251);

//尾巴

x0[19]=k\*(x+282);

y0[19]=k\*(y+233);

x0[20]=k\*(x+280);

y0[20]=k\*(y+224);

x0[21]=k\*(x+278);

y0[21]=k\*(y+228);

x0[22]=k\*(x+272);

y0[22]=k\*(y+216);

x0[23]=k\*(x+286);

y0[23]=k\*(y+211);

x0[24]=k\*(x+294);

y0[24]=k\*(y+209);

x0[25]=k\*(x+298);

y0[25]=k\*(y+205);

x0[26]=k\*(x+299);

y0[26]=k\*(y+204);

x0[27]=k\*(x+302);

y0[27]=k\*(y+200);

x0[28]=k\*(x+304);

y0[28]=k\*(y+197);

x0[29]=k\*(x+303);

y0[29]=k\*(y+194);

x0[30]=k\*(x+300);

y0[30]=k\*(y+191);

x0[31]=k\*(x+298);

y0[31]=k\*(y+190);

x0[32]=k\*(x+296);

y0[32]=k\*(y+189);

x0[33]=k\*(x+294);

y0[33]=k\*(y+188);

x0[34]=k\*(x+292);

y0[34]=k\*(y+187);

x0[35]=k\*(x+289);

y0[35]=k\*(y+186);

x0[36]=k\*(x+287);

y0[36]=k\*(y+186);

x0[37]=k\*(x+285);

y0[37]=k\*(y+185);

x0[38]=k\*(x+283);

y0[38]=k\*(y+184);

x0[39]=k\*(x+282);

y0[39]=k\*(y+185);

x0[40]=k\*(x+279);

y0[40]=k\*(y+186);

x0[41]=k\*(x+277);

y0[41]=k\*(y+188);

x0[42]=k\*(x+276);

y0[42]=k\*(y+190);

x0[43]=k\*(x+274);

y0[43]=k\*(y+192);

x0[44]=k\*(x+263);

y0[44]=k\*(y+194);

x0[45]=k\*(x+261);

y0[45]=k\*(y+198);

x0[46]=k\*(x+263);

y0[46]=k\*(y+201);

x0[47]=k\*(x+264);

y0[47]=k\*(y+208);

x0[48]=k\*(x+267);

y0[48]=k\*(y+212);

x0[49]=k\*(x+267);

y0[49]=k\*(y+219);

x0[50]=k\*(x+268);

y0[50]=k\*(y+227);

x0[51]=k\*(x+269);

y0[51]=k\*(y+231);

x0[52]=k\*(x+272);

y0[52]=k\*(y+238);

x0[53]=k\*(x+273);

y0[53]=k\*(y+241);

x0[54]=k\*(x+275);

y0[54]=k\*(y+245);

x0[55]=k\*(x+277);

y0[55]=k\*(y+251);

x0[56]=k\*(x+280);

y0[56]=k\*(y+255);

x0[57]=k\*(x+286);

y0[57]=k\*(y+258);

x0[58]=k\*(x+284);

y0[58]=k\*(y+260);

x0[59]=k\*(x+283);

y0[59]=k\*(y+264);

x0[60]=k\*(x+275);

y0[60]=k\*(y+268);

x0[61]=k\*(x+273);

y0[61]=k\*(y+268);

x0[62]=k\*(x+272);

y0[62]=k\*(y+268);

x0[63]=k\*(x+270);

y0[63]=k\*(y+275);

x0[64]=k\*(x+265);

y0[64]=k\*(y+279);

x0[65]=k\*(x+245);

y0[65]=k\*(y+281);

x0[66]=k\*(x+245);

y0[66]=k\*(y+283);

x0[67]=k\*(x+240);

y0[67]=k\*(y+305);

x0[68]=k\*(x+239);

y0[68]=k\*(y+308);

x0[69]=k\*(x+238);

y0[69]=k\*(y+311);

x0[70]=k\*(x+237);

y0[70]=k\*(y+315);

x0[71]=k\*(x+237);

y0[71]=k\*(y+321);

x0[72]=k\*(x+236);

y0[72]=k\*(y+330);

x0[73]=k\*(x+234);

y0[73]=k\*(y+338);

x0[74]=k\*(x+233);

y0[74]=k\*(y+344);

x0[75]=k\*(x+232);

y0[75]=k\*(y+350);

x0[76]=k\*(x+232);

y0[76]=k\*(y+355);

x0[77]=k\*(x+233);

y0[77]=k\*(y+368);

x0[78]=k\*(x+235);

y0[78]=k\*(y+375);

x0[79]=k\*(x+239);

y0[79]=k\*(y+385);

x0[80]=k\*(x+240);

y0[80]=k\*(y+389);

x0[81]=k\*(x+243);

y0[81]=k\*(y+393);

x0[82]=k\*(x+245);

y0[82]=k\*(y+394);

x0[83]=k\*(x+248);

y0[83]=k\*(y+396);

x0[84]=k\*(x+252);

y0[84]=k\*(y+391);

x0[85]=k\*(x+254);

y0[85]=k\*(y+387);

x0[86]=k\*(x+256);

y0[86]=k\*(y+383);

x0[87]=k\*(x+257);

y0[87]=k\*(y+379);

x0[88]=k\*(x+259);

y0[88]=k\*(y+373);

x0[89]=k\*(x+259);

y0[89]=k\*(y+369);

x0[90]=k\*(x+260);

y0[90]=k\*(y+363);

x0[91]=k\*(x+262);

y0[91]=k\*(y+359);

x0[92]=k\*(x+264);

y0[92]=k\*(y+355);

x0[93]=k\*(x+265);

y0[93]=k\*(y+346);

x0[94]=k\*(x+268);

y0[94]=k\*(y+343);

x0[95]=k\*(x+270);

y0[95]=k\*(y+340);

x0[96]=k\*(x+273);

y0[96]=k\*(y+335);

x0[97]=k\*(x+276);

y0[97]=k\*(y+238);

x0[98]=k\*(x+280);

y0[98]=k\*(y+300);

x0[99]=k\*(x+283);

y0[99]=k\*(y+280);

x0[100]=k\*(x+285);

y0[100]=k\*(y+282);

x0[101]=k\*(x+288);

y0[101]=k\*(y+300);

x0[102]=k\*(x+293);

y0[102]=k\*(y+302);

//左后腿

x0[103]=k\*(x+292);

y0[103]=k\*(y+314);

x0[104]=k\*(x+290);

y0[104]=k\*(y+318);

x0[105]=k\*(x+289);

y0[105]=k\*(y+312);

x0[106]=k\*(x+287);

y0[106]=k\*(y+316);

x0[107]=k\*(x+286);

y0[107]=k\*(y+328);

x0[108]=k\*(x+285);

y0[108]=k\*(y+324);

x0[109]=k\*(x+284);

y0[109]=k\*(y+339);

x0[110]=k\*(x+285);

y0[110]=k\*(y+352);

x0[111]=k\*(x+285);

y0[111]=k\*(y+353);

x0[112]=k\*(x+283);

y0[112]=k\*(y+354);

x0[113]=k\*(x+283);

y0[113]=k\*(y+355);

x0[114]=k\*(x+283);

y0[114]=k\*(y+356);

x0[115]=k\*(x+281);

y0[115]=k\*(y+356);

x0[116]=k\*(x+280);

y0[116]=k\*(y+358);

x0[117]=k\*(x+278);

y0[117]=k\*(y+350);

x0[118]=k\*(x+278);

y0[118]=k\*(y+342);

x0[119]=k\*(x+279);

y0[119]=k\*(y+345);

x0[120]=k\*(x+280);

y0[120]=k\*(y+349);

x0[121]=k\*(x+281);

y0[121]=k\*(y+349);

x0[122]=k\*(x+279);

y0[122]=k\*(y+352);

x0[123]=k\*(x+280);

y0[123]=k\*(y+355);

x0[124]=k\*(x+280);

y0[124]=k\*(y+358);

x0[125]=k\*(x+281);

y0[125]=k+(y+360);

x0[126]=k\*(x+283);

y0[126]=k\*(y+362);

x0[127]=k\*(x+284);

y0[127]=k\*(y+365);

x0[128]=k\*(x+286);

y0[128]=k\*(y+366);

x0[129]=k\*(x+287);

y0[129]=k\*(y+367);

x0[130]=k\*(x+289);

y0[130]=k\*(y+366);

x0[131]=k\*(x+289);

y0[131]=k\*(y+365);

x0[132]=k\*(x+289);

y0[132]=k\*(y+366);

x0[133]=k\*(x+291);

y0[133]=k\*(y+367);

x0[134]=k\*(x+292);

y0[134]=k\*(y+367);

x0[135]=k\*(x+293);

y0[135]=k\*(y+365);

x0[136]=k\*(x+293);

y0[136]=k\*(y+364);

x0[137]=k\*(x+293);

y0[137]=k\*(y+363);

x0[138]=k\*(x+293);

y0[138]=k\*(y+363);

x0[139]=k\*(x+294);

y0[139]=k\*(y+363);

x0[140]=k\*(x+296);

y0[140]=k\*(y+361);

x0[141]=k\*(x+297);

y0[141]=k\*(y+357);

x0[142]=k\*(x+297);

y0[142]=k\*(y+354);

x0[143]=k\*(x+297);

y0[143]=k\*(y+354);

x0[144]=k\*(x+298);

y0[144]=k\*(y+351);

x0[145]=k\*(x+298);

y0[145]=k\*(y+348);

x0[146]=k\*(x+298);

y0[146]=k\*(y+345);

x0[147]=k\*(x+299);

y0[147]=k\*(y+343);

x0[148]=k\*(x+300);

y0[148]=k\*(y+341);

x0[149]=k\*(x+301);

y0[149]=k\*(y+339);

x0[150]=k\*(x+303);

y0[150]=k\*(y+334);

x0[151]=k\*(x+304);

y0[151]=k\*(y+329);

x0[152]=k\*(x+306);

y0[152]=k\*(y+330);

x0[153]=k\*(x+308);

y0[153]=k\*(y+331);

x0[154]=k\*(x+309);

y0[154]=k\*(y+331);

x0[155]=k\*(x+310);

y0[155]=k\*(y+332);

x0[156]=k\*(x+312);

y0[156]=k\*(y+333);

x0[157]=k\*(x+313);

y0[157]=k\*(y+333);

x0[158]=k\*(x+316);

y0[158]=k\*(y+336);

x0[159]=k\*(x+318);

y0[159]=k\*(y+336);

x0[160]=k\*(x+322);

y0[160]=k\*(y+337);

x0[161]=k\*(x+324);

y0[161]=k\*(y+338);

x0[162]=k\*(x+326);

y0[162]=k\*(y+338);

x0[163]=k\*(x+330);

y0[163]=k\*(y+337);

x0[164]=k\*(x+336);

y0[164]=k\*(y+336);

x0[165]=k\*(x+339);

y0[165]=k\*(y+335);

x0[166]=k\*(x+343);

y0[166]=k\*(y+334);

x0[167]=k\*(x+346);

y0[167]=k\*(y+333);

x0[168]=k\*(x+349);

y0[168]=k\*(y+329);

x0[169]=k\*(x+354);

y0[169]=k\*(y+328);

x0[170]=k\*(x+357);

y0[170]=k\*(y+328);

x0[171]=k\*(x+362);

y0[171]=k\*(y+327);

x0[172]=k\*(x+363);

y0[172]=k\*(y+325);

x0[173]=k\*(x+366);

y0[173]=k\*(y+325);

//右前腿

for(i=0;i<38;i++)

{

x0[174+i]=k\*(x+366);

y0[174+i]=k\*(y+326+i);

}

x0[212]=k\*(x+368);

y0[212]=k\*(y+363);

x0[213]=k\*(x+370);

y0[213]=k\*(y+368);

x0[214]=k\*(x+377);

y0[214]=k\*(y+372);

x0[215]=k\*(x+381);

y0[215]=k\*(y+374);

x0[216]=k\*(x+386);

y0[216]=k\*(y+373);

x0[217]=k\*(x+389);

y0[217]=k\*(y+373);

x0[218]=k\*(x+387);

y0[218]=k\*(y+373);

x0[219]=k\*(x+386);

y0[219]=k\*(y+373);

x0[220]=k\*(x+387);

y0[220]=k\*(y+370);

x0[221]=k\*(x+388);

y0[221]=k\*(y+367);

x0[222]=k\*(x+389);

y0[222]=k\*(y+360);

x0[223]=k\*(x+391);

y0[223]=k\*(y+324);

x0[224]=k\*(x+393);

y0[224]=k\*(y+315);

x0[225]=k\*(x+395);

y0[225]=k\*(y+315);

x0[226]=k\*(x+397);

y0[226]=k\*(y+315);

x0[227]=k\*(x+399);

y0[227]=k\*(y+315);

x0[228]=k\*(x+402);

y0[228]=k\*(y+314);

x0[229]=k\*(x+405);

y0[229]=k\*(y+313);

x0[230]=k\*(x+409);

y0[230]=k\*(y+322);

x0[231]=k\*(x+416);

y0[231]=k\*(y+321);

for(i=0;i<50;i++)

{

x0[232+i]=k\*(x+416);

y0[232+i]=k\*(y+321+i);

}

x0[282]=k\*(x+425);

y0[282]=k\*(y+371);

x0[283]=k\*(x+430);

y0[283]=k\*(y+374);

x0[284]=k\*(x+434);

y0[284]=k\*(y+372);

x0[285]=k\*(x+435);

y0[285]=k\*(y+372);

x0[286]=k\*(x+441);

y0[286]=k\*(y+373);

for(i=0;i<60;i++)

{

x0[287+i]=k\*(x+441);

y0[287+i]=k\*(y+373-i);

}

x0[347]=k\*(x+443);

y0[347]=k\*(y+310);

x0[348]=k\*(x+448);

y0[348]=k\*(y+300);

x0[349]=k\*(x+453);

y0[349]=k\*(y+290);

x0[350]=k\*(x+458);

y0[350]=k\*(y+280);

x0[351]=k\*(x+461);

y0[351]=k\*(y+277);

x0[352]=k\*(x+463);

y0[352]=k\*(y+281);

x0[353]=k\*(x+465);

y0[353]=k\*(y+284);

x0[354]=k\*(x+469);

y0[354]=k\*(y+284);

x0[355]=k\*(x+472);

y0[355]=k\*(y+286);

x0[356]=k\*(x+476);

y0[356]=k\*(y+289);

x0[357]=k\*(x+481);

y0[357]=k\*(y+291);

x0[358]=k\*(x+485);

y0[358]=k\*(y+295);

x0[359]=k\*(x+494);

y0[359]=k\*(y+300);

x0[360]=k\*(x+500);

y0[360]=k\*(y+304);

x0[361]=k\*(x+503);

y0[361]=k\*(y+299);

x0[362]=k\*(x+504);

y0[362]=k\*(y+295);

x0[363]=k\*(x+507);

y0[363]=k\*(y+291);

x0[364]=k\*(x+508);

y0[364]=k\*(y+288);

x0[365]=k\*(x+510);

y0[365]=k\*(y+284);

x0[366]=k\*(x+510);

y0[366]=k\*(y+282);

x0[367]=k\*(x+513);

y0[367]=k\*(y+279);

x0[368]=k\*(x+515);

y0[368]=k\*(y+275);

x0[369]=k\*(x+519);

y0[369]=k\*(y+271);

x0[370]=k\*(x+521);

y0[370]=k\*(y+261);

x0[371]=k\*(x+525);

y0[371]=k\*(y+247);

x0[372]=k\*(x+529);

y0[372]=k\*(y+242);

x0[373]=k\*(x+531);

y0[373]=k\*(y+236);

x0[374]=k\*(x+535);

y0[374]=k\*(y+230);

x0[375]=k\*(x+528);

y0[375]=k\*(y+227);

x0[376]=k\*(x+526);

y0[376]=k\*(y+226);

x0[377]=k\*(x+522);

y0[377]=k\*(y+225);

x0[378]=k\*(x+519);

y0[378]=k\*(y+226);

x0[379]=k\*(x+514);

y0[379]=k\*(y+223);

x0[380]=k\*(x+512);

y0[380]=k\*(y+221);

x0[381]=k\*(x+509);

y0[381]=k\*(y+221);

x0[382]=k\*(x+506);

y0[382]=k\*(y+221);

x0[383]=k\*(x+498);

y0[383]=k\*(y+221);

x0[384]=k\*(x+491);

y0[384]=k\*(y+220);

x0[385]=k\*(x+486);

y0[385]=k\*(y+219);

x0[386]=k\*(x+483);

y0[386]=k\*(y+219);

x0[387]=k\*(x+481);

y0[387]=k\*(y+218);

x0[388]=k\*(x+477);

y0[388]=k\*(y+215);

x0[389]=k\*(x+472);

y0[389]=k\*(y+215);

x0[390]=k\*(x+470);

y0[390]=k\*(y+211);

x0[391]=k\*(x+467);

y0[391]=k\*(y+208);

x0[392]=k\*(x+464);

y0[392]=k\*(y+206);

x0[393]=k\*(x+461);

y0[393]=k\*(y+204);

x0[394]=k\*(x+459);

y0[394]=k\*(y+202);

x0[395]=k\*(x+456);

y0[395]=k\*(y+199);

x0[396]=k\*(x+451);

y0[396]=k\*(y+198);

x0[397]=k\*(x+448);

y0[397]=k\*(y+196);

x0[398]=k\*(x+437);

y0[398]=k\*(y+190);

x0[399]=k\*(x+435);

y0[399]=k\*(y+185);

x0[400]=k\*(x+434);

y0[400]=k\*(y+186);

x0[401]=k\*(x+432);

y0[401]=k\*(y+186);

x0[402]=k\*(x+430);

y0[402]=k\*(y+186);

x0[403]=k\*(x+428);

y0[403]=k\*(y+186);

x0[404]=k\*(x+427);

y0[404]=k\*(y+185);

x0[405]=k\*(x+421);

y0[405]=k\*(y+184);

x0[406]=k\*(x+415);

y0[406]=k\*(y+183);

x0[407]=k\*(x+412);

y0[407]=k\*(y+184);

x0[408]=k\*(x+408);

y0[408]=k\*(y+183);

x0[409]=k\*(x+395);

y0[409]=k\*(y+183);

x0[410]=k\*(x+391);

y0[410]=k\*(y+182);

x0[411]=k\*(x+387);

y0[411]=k\*(y+181);

x0[412]=k\*(x+385);

y0[412]=k\*(y+185);

x0[413]=k\*(x+385);

y0[413]=k\*(y+186);

x0[414]=k\*(x+385);

y0[414]=k\*(y+188);

x0[415]=k\*(x+385);

y0[415]=k\*(y+190);

x0[416]=k\*(x+385);

y0[416]=k\*(y+192);

x0[417]=k\*(x+385);

y0[417]=k\*(y+194);

x0[418]=k\*(x+385);

y0[418]=k\*(y+197);

x0[419]=k\*(x+385);

y0[419]=k\*(y+201);

x0[420]=k\*(x+385);

y0[420]=k\*(y+204);

x0[421]=k\*(x+386);

y0[421]=k\*(y+210);

x0[422]=k\*(x+386);

y0[422]=k\*(y+215);

x0[423]=k\*(x+386);

y0[423]=k\*(y+220);

x0[424]=k\*(x+386);

y0[424]=k\*(y+224);

x0[425]=k\*(x+390);

y0[425]=k\*(y+229);

x0[426]=k\*(x+392);

y0[426]=k\*(y+233);

x0[427]=k\*(x+395);

y0[427]=k\*(y+234);

x0[428]=k\*(x+410);

y0[428]=k\*(y+235);

x0[429]=k\*(x+414);

y0[429]=k\*(y+238);

x0[430]=k\*(x+414);

y0[430]=k\*(y+244);

x0[431]=k\*(x+414);

y0[431]=k\*(y+245);

//把初始顶点复原

for(i=0;i<432;i++)

{

x0[i]=x0[i]-(x+414)\*(k-1);

y0[i]=y0[i]-(y+245)\*(k-1);

}

//把限制位置的坐标复原

posi[0]=posi[0]-(x+414)\*(k-1);

posi[1]=posi[1]-(x+414)\*(k-1);

posi[2]=posi[2]-(y+245)\*(k-1);

posi[3]=posi[3]-(y+245)\*(k-1);

// circle\_with\_drawing(300,300,20,BLUE\_,page,1);

point\_drawing(k\*(x+414),k\*(y+246),color,page);

Bezier(color,432,page,x0,y0,0);

fill\_1(color,page);

circle\_with\_drawing(k\*(x+458),k\*(y+234),8,BLUE\_,page,1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 函数名称：SVGA基本逻辑

\*\* 函数目的：实现SVGA

\*\* 时间复杂度：公认的最快速度

\*\* 空间复杂度：最小空间

\*/

#include<stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<xiezi.h>

#include<mouse.h>

#include<draw.h>

//函数名称：start\_SVGA

//函数目的：初始化65536色SVGA显示

void start\_SVGA()

{

union REGS our\_AI\_regs;

our\_AI\_regs.x.ax=0x4f02;//设置显示模式

our\_AI\_regs.x.bx=0x116;//设置成65536色SVGA显示

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);

//软中断出现SVGA特权功能，寄存器存储值

if(our\_AI\_regs.x.ax!=0x004f)

{

printf("初始化失败！错误：0x%x！任意键退出",our\_AI\_regs.x.ax);

getchar();

exit(1);

}

}

//函数名称：examine\_start\_SVGA

//函数目的：判断是否初始化成功

int examine\_start\_SVGA()

{

union REGS our\_AI\_regs;

our\_AI\_regs.x.ax=0x4f03;//读当前显示模式

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);//软中断

if(our\_AI\_regs.x.ax!=0x004f)

{

printf("初始化失败！错误：0x%x！任意键退出",our\_AI\_regs.x.ax);

getchar();

exit(1);

}

return 1;

}

//函数名称：pageturn

//函数目的：换页问题,换到第page页

/\*

void pageturn(register int page)//用寄存器，加快换页速度

{

static int flag = 0; // 0 代表 false，1 代表 true,BC不支持布尔值

union REGS our\_AI\_regs;

static int whether=0;//这个不能变

static int initial\_page;//这个也需要保持不变

our\_AI\_regs.x.ax=0x4f05;//控制内存页区域切换

our\_AI\_regs.x.bx=0;//寄存器设初值为0

if(whether==0)

{

our\_AI\_regs.x.dx=page;//换页

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);//软中断

flag=1;

initial\_page=page;//数据维护

}

else if(page!=initial\_page)

{

our\_AI\_regs.x.dx=page;

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);

initial\_page=page;

}

}

\*/

void pageturn(register int page)

{

union REGS our\_AI\_regs;

our\_AI\_regs.x.ax=0x4f05;

our\_AI\_regs.x.bx=0;

our\_AI\_regs.x.dx=page;

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);

}

//函数名称：point\_drawing

//函数目的：由点出线，确定画点函数

int point\_drawing(int x,int y,int color,int page)

{

unsigned int far \*const buffer= (unsigned int far\*)0xa0000000L;

//指针首地址，可读不可写可执行

if(x<0||x>XMAX||y<0||y>YMAX)//自己要先定义一下！！！

{

return 0;

}

//如果点不在屏幕里面，就不用再算了

unsigned long long int delta=((unsigned long long int)y<<10)+x;

//计算像素偏移量delta，1024=2<<10,一个y就是这么多像素

int i=y/32;

int eventual\_page=(page>>16)+i;//2^16个点换页一次

pageturn(eventual\_page);

buffer[delta]=color;

}

//函数名称：line\_drawing

//函数目的：用Bresenham算法画线

int line\_drawing(int x1,int y1,int x2,int y2,int color,int page,int status)

{

int delta\_x=abs(x2-x1);

int delta\_y=abs(y2-y1);//计算差值

int delta\_x2,delta\_y2;

int sign\_x=0,sign\_y=0;

int i,min,max;

int error,flag;

if(status==0)

{

if(x1<x2)

{

sign\_x=1;

}

else if(x1>x2)

{

sign\_x=-1;

}

else//竖直线段直接画

{

if(y1>y2)

{

max=y1;

min=y2;

}

else

{

max=y2;

min=y1;

}

for(i=min;i<=max;i++)

{

point\_drawing(x1,i,color,page);

}

//return 0;

}

if(y1<y2)

{

sign\_y=1;

}

else if(y1>y2)

{

sign\_y=-1;

}

else//水平直线直接画

{

if(x1<x2)

{

max=x2;

min=x1;

}

else

{

max=x1;

min=x2;

}

for(i=min;i<=max;i++)

{

point\_drawing(i,y1,color,page);

}

}

point\_drawing(x1,y1,color,page);

delta\_x2=delta\_x<<1;

delta\_y2=delta\_y<<1;//方便后续2倍运算

if(delta\_x>=delta\_y)//横坐标适合作为基准

{

error=delta\_y2-delta\_x;//确定误差项

flag=delta\_y2-delta\_x2;

while(delta\_x--)//两边x在靠近，要是到0就不执行了

{

if(error<=0)

{

error+=delta\_y2;

}

else

{

y1+=sign\_y;

error+=flag;

}

//平均每走delta\_x2-delta\_y2/delta\_y2+1 个格就画一个点,那个+1就是else里面没有执行可是x还动了一步

x1+=sign\_x;

point\_drawing(x1,y1,color,page);

}

}

else

{

error=delta\_x2-delta\_y;

flag=delta\_x2-delta\_y2;

while(delta\_y--)

{

if(error<=0)

{

error+=delta\_x2;

}

else

{

x1+=sign\_x;

error+=flag;

}

y1+=sign\_y;

point\_drawing(x1,y1,color,page);

}

}

}

else if(status==1)

{

if(x1<x2)

{

sign\_x=1;

}

else if(x1>x2)

{

sign\_x=-1;

}

else//竖直线段直接画

{

if(y1>y2)

{

max=y1;

min=y2;

}

else

{

max=y2;

min=y1;

}

for(i=min;i<=max;i++)

{

if(i<=565)

{

point\_drawing(x1,i,color,page);

}

}

//return 0;

}

if(y1<y2)

{

sign\_y=1;

}

else if(y1>y2)

{

sign\_y=-1;

}

else//水平直线直接画

{

if(x1<x2)

{

max=x2;

min=x1;

}

else

{

max=x1;

min=x2;

}

for(i=min;i<=max;i++)

{

if(y1<=565)

{

point\_drawing(i,y1,color,page);

}

}

}

if(y1<=565)

{

point\_drawing(x1,y1,color,page);

}

delta\_x2=delta\_x<<1;

delta\_y2=delta\_y<<1;//方便后续2倍运算

if(delta\_x>=delta\_y)//横坐标适合作为基准

{

error=delta\_y2-delta\_x;//确定误差项

flag=delta\_y2-delta\_x2;

while(delta\_x--)//两边x在靠近，要是到0就不执行了

{

if(error<=0)

{

error+=delta\_y2;

}

else

{

y1+=sign\_y;

error+=flag;

}

//平均每走delta\_x2-delta\_y2/delta\_y2+1 个格就画一个点,那个+1就是else里面没有执行可是x还动了一步

x1+=sign\_x;

if(y1<=565)

{

point\_drawing(x1,y1,color,page);

}

}

}

else

{

error=delta\_x2-delta\_y;

flag=delta\_x2-delta\_y2;

while(delta\_y--)

{

if(error<=0)

{

error+=delta\_x2;

}

else

{

x1+=sign\_x;

error+=flag;

}

y1+=sign\_y;

if(y1<=565)

{

point\_drawing(x1,y1,color,page);

}

}

}

}

}

//函数名称：circle\_without\_drawing

//函数目的：Bresenham算法画圆圈

void circle\_without\_drawing(int x,int y,int r,int color,int page)

{

if(r<=0)

{

printf("半径为负！任意键退出");

getchar();

exit(1);

}

int i=r;

int error=3-2\*r;//误差项

int j;

for(j=0;j<=i;j++)

{

point\_drawing(x+j,y+i,color,page);

point\_drawing(x+j,y-i,color,page);

point\_drawing(x-j,y+i,color,page);

point\_drawing(x-j,y-i,color,page);

point\_drawing(x+i,y+j,color,page);

point\_drawing(x+i,y-j,color,page);

point\_drawing(x-i,y+j,color,page);

point\_drawing(x-i,y-j,color,page);

//圆的对称性画8个对称的点

if(error<0)

{

error+=4\*j+6;

}

else

{

error+=4\*(j-i)+10;

i--;

}

}

}

//函数名称：circle\_with\_drawing

//函数目的：画实心圆

void circle\_with\_drawing(int x,int y,int r,int color,int page,int status)

{

int setx=0,sety=r,dx=3,d2=2-2\*r,d=1-r;

if(r<=0)

{

printf("半径为负！任意键退出");

getchar();

exit(1);

}

while(setx<=sety)

{

line\_drawing(x-setx,y-sety,x+setx,y-sety,color,page,status);

line\_drawing(x-sety,y-setx,x+sety,y-setx,color,page,status);

line\_drawing(x-sety,y+setx,x+sety,y+setx,color,page,status);

line\_drawing(x-setx,y+sety,x+setx,y+sety,color,page,status);

if(d<0)

{

d+=dx;

dx+=2;

}

else

{

sety--;

d+=(dx+d2);

dx+=2;

d2+=2;

}

setx++;

}

}

//函数名称：fill\_color

//函数目的：实现颜色填充矩形画面

void fill\_color(int x1,int y1,int x2,int y2,int color,int page,int status)

{

int i,j;

if(status==0)

{

for(i=x1;i<=x2;i++)

{

for(j=y1;j<=y2;j++)

{

point\_drawing(i,j,color,page);

}

}

}

else

{

if(y1<=575&&y2>=575)

{

for(i=x1;i<=x2;i++)

{

for(j=y1;j<=575;j++)

{

point\_drawing(i,j,color,page);

}

}

}

if(y1<=575&&y2<=575)

{

for(i=x1;i<=x2;i++)

{

for(j=y1;j<=y2;j++)

{

point\_drawing(i,j,color,page);

}

}

}

}

}

//函数名称：exit\_SVGA

//函数目的：关闭SVGA

void exit\_SVGA()

{

union REGS our\_AI\_regs;

our\_AI\_regs.h.ah=0;

our\_AI\_regs.h.al=0x03;

int86(0x10,&our\_AI\_regs,&our\_AI\_regs);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<English.h>

#include<stdio.h>

#include<string.h>

#include<draw.h>

#include<stdlib.h>

#include<alloc.h>

void English\_drawing(int x,int y,char c,int color,int page,int status)

{

//unsigned long long int\*\* ascii\_font=(unsigned long long int\*\*)malloc(95\*sizeof(unsigned long long int\*));

//int i;

//for(i=0;i<95;i++)

//{

// ascii\_font[i]=(unsigned long long int\*)malloc(33\*sizeof(unsigned long long int));

//}

unsigned long long int ascii\_font[95][32]=

{

//' '

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'!

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00},

//'"

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x03CF00, 0x03CF00, 0x03CF00, 0x03CF00, 0x03CF00, 0x03CF00, 0x03CF00, 0x03CF00, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'#

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0078E0, 0x0071E0, 0x00F1E0, 0x00F1C0, 0x00F1C0, 0x1FFFFC, 0x1FFFFC, 0x1FFFFC, 0x01E380, 0x01C780, 0x03C780, 0x03C780, 0x3FFFF0, 0x3FFFF0, 0x3FFFF0, 0x078F00, 0x078E00, 0x071E00, 0x071E00, 0x0F1C00},

//'$

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x003800, 0x003800, 0x003800, 0x01FFE0, 0x03FFE0, 0x07FFE0, 0x0FF9E0, 0x0FB800, 0x0FB800, 0x0FF800, 0x0FFF00, 0x07FFC0, 0x03FFE0, 0x01FFE0, 0x003FF0, 0x0039F0, 0x0039F0, 0x0F3BF0, 0x0FFFE0, 0x0FFFC0, 0x0FFF80, 0x003800, 0x003800, 0x003800},

//'%

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x7E00F0, 0xFF80E0, 0xF781E0, 0xE3C3C0, 0xE3C380, 0xE3C780, 0xE3CF00, 0xE3CE00, 0xF79E00, 0xFF9C7E, 0x7E3DFF, 0x0079EF, 0x0073C7, 0x00F3C7, 0x01E3C7, 0x01C3C7, 0x03C3C7, 0x0781EF, 0x0701FF, 0x0F007E},

//'&

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x03FF80, 0x07FF80, 0x07E780, 0x0FC180, 0x07C000, 0x07E000, 0x07F000, 0x0FF83E, 0x1FFC3C, 0x3FFE7C, 0x3E7F7C, 0x3E3FFC, 0x7E1FF8, 0x7E0FF0, 0x3F07F0, 0x3F8FF0, 0x1FFFF8, 0x0FFFFC, 0x03FE7E},

//'\

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007800, 0x007800, 0x007800, 0x007800, 0x007800, 0x007800, 0x007800, 0x007800, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'(

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x001F00, 0x003E00, 0x003E00, 0x007C00, 0x007C00, 0x00FC00, 0x00F800, 0x00F800, 0x00F800, 0x01F800, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F800, 0x00F800, 0x00F800, 0x00F800, 0x00FC00, 0x007C00, 0x007C00, 0x003E00, 0x003E00},

//')

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01F000, 0x00F000, 0x00F800, 0x007800, 0x007C00, 0x007C00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x007C00, 0x007C00, 0x007800, 0x00F800, 0x00F000},

//'\*

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x003800, 0x003800, 0x0638C0, 0x07BBC0, 0x07FFC0, 0x01FF00, 0x00FC00, 0x01FF00, 0x07FFC0, 0x07BBC0, 0x0638C0, 0x003800, 0x003800, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'+

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00},

//',

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x0000F0, 0x0000F0, 0x003FF0, 0x00FFE0, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'-

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x01FF00, 0x01FF00, 0x01FF00, 0x000000},

//'.

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x000FF0, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'/

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000700, 0x000F00, 0x000F00, 0x000E00, 0x001E00, 0x001E00, 0x001C00, 0x001C00, 0x003C00, 0x003C00, 0x003800, 0x007800, 0x007800, 0x007000, 0x00F000, 0x00F000, 0x00E000, 0x01E000, 0x01E000, 0x01C000, 0x03C000, 0x03C000},

//'0

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FF00, 0x03FF80, 0x07FFC0, 0x07FFE0, 0x0FE7E0, 0x0FC3F0, 0x1F83F0, 0x1F83F0, 0x1F81F0, 0x1F81F0, 0x1F81F0, 0x1F81F0, 0x1F83F0, 0x1F83F0, 0x0FC3F0, 0x0FE7E0, 0x07FFE0, 0x07FFC0, 0x03FF80, 0x00FF00},

//'1

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x03FE00, 0x07FE00, 0x07FE00, 0x07FE00, 0x07BE00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x003E00, 0x07FFF0, 0x07FFF0, 0x07FFF0, 0x07FFF0},

//'2

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07FF00, 0x0FFF80, 0x0FFFC0, 0x0FFFE0, 0x0F0FE0, 0x0007E0, 0x0007E0, 0x0007E0, 0x000FC0, 0x001FC0, 0x003F80, 0x007F00, 0x00FC00, 0x01F800, 0x03F000, 0x0FE000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0},

//'3

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x03FF00, 0x0FFF80, 0x0FFFC0, 0x0FFFE0, 0x0F0FE0, 0x0007E0, 0x0007E0, 0x000FC0, 0x01FF80, 0x01FF00, 0x01FFC0, 0x01FFE0, 0x000FE0, 0x0003E0, 0x0003E0, 0x0E0FE0, 0x0FFFE0, 0x0FFFC0, 0x0FFF80, 0x07FE00},

//'4

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x001FC0, 0x003FC0, 0x007FC0, 0x007FC0, 0x00FFC0, 0x01FFC0, 0x01EFC0, 0x03CFC0, 0x078FC0, 0x0F8FC0, 0x0F0FC0, 0x1E0FC0, 0x1FFFF0, 0x1FFFF0, 0x1FFFF0, 0x1FFFF0, 0x000FC0, 0x000FC0, 0x000FC0, 0x000FC0},

//'5

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07FFC0, 0x07FFC0, 0x07FFC0, 0x07FFC0, 0x078000, 0x078000, 0x07FF00, 0x07FF80, 0x07FFC0, 0x07FFE0, 0x070FE0, 0x0007F0, 0x0003F0, 0x0003F0, 0x0007F0, 0x0F0FE0, 0x0FFFE0, 0x0FFFC0, 0x0FFF80, 0x07FE00},

//'6

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007FC0, 0x01FFE0, 0x03FFE0, 0x07FFE0, 0x07F0E0, 0x0FC000, 0x0FBF00, 0x0FFFC0, 0x0FFFE0, 0x0FFFF0, 0x1FE7F0, 0x0FC3F0, 0x0FC1F0, 0x0FC1F0, 0x0FC3F0, 0x0FE7F0, 0x07FFE0, 0x03FFE0, 0x01FFC0, 0x00FF00},

//'7

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FFFF0, 0x0FFFF0, 0x0FFFF0, 0x0FFFE0, 0x0007E0, 0x0007C0, 0x000FC0, 0x000F80, 0x001F80, 0x001F00, 0x003F00, 0x003F00, 0x007E00, 0x007E00, 0x00FC00, 0x00FC00, 0x00F800, 0x01F800, 0x01F000, 0x03F000},

//'8

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x03FFC0, 0x07FFE0, 0x0FFFE0, 0x0FE7F0, 0x0FC3F0, 0x0FC3E0, 0x0FC7E0, 0x07FFC0, 0x03FF80, 0x03FFC0, 0x07FFE0, 0x0FC7F0, 0x0F83F0, 0x1F83F0, 0x0FC7F0, 0x0FFFF0, 0x0FFFE0, 0x07FFC0, 0x01FF80},

//'9

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FE00, 0x03FF80, 0x07FFC0, 0x0FFFC0, 0x0FCFE0, 0x1F87E0, 0x1F83F0, 0x1F83F0, 0x1F87F0, 0x1FCFF0, 0x0FFFF0, 0x0FFFF0, 0x07FFF0, 0x01FBE0, 0x0007E0, 0x071FE0, 0x07FFC0, 0x07FF80, 0x07FF00, 0x03FC00},

//':

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00},

//';

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00},

//'<

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000038, 0x0000F8, 0x0007F8, 0x003FF8, 0x01FFC0, 0x07FE00, 0x1FF000, 0x1FC000, 0x1FC000, 0x1FF000, 0x07FE00, 0x01FFC0, 0x003FF8, 0x0007F8, 0x0000F8, 0x000038},

//'=

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x000000, 0x000000, 0x000000, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x000000},

//'>

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x180000, 0x1F0000, 0x1FE000, 0x1FF800, 0x07FF00, 0x00FFE0, 0x001FF8, 0x0003F8, 0x0003F8, 0x001FF8, 0x00FFE0, 0x07FF00, 0x1FF800, 0x1FE000, 0x1F0000, 0x180000},

//'?

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x03FF00, 0x03FF80, 0x03FFC0, 0x038FE0, 0x0007E0, 0x0007E0, 0x000FC0, 0x001FC0, 0x003F80, 0x007F00, 0x007E00, 0x007C00, 0x00FC00, 0x000000, 0x000000, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00},

//'@

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FF80, 0x03FFE0, 0x0FFFF0, 0x1FC3F8, 0x3E007C, 0x3C001E, 0x787DCE, 0x70FFCF, 0xF1FFC7, 0xE3E7C7, 0xE3C3C7, 0xE3C3C7, 0xE3C3C7, 0xE3C3C7, 0xE3C3CF, 0xE3E7DE, 0xF1FFFC, 0x70FFF8, 0x787DF0, 0x3C0000, 0x3E0060, 0x1FC3F0, 0x0FFFF0},

//'A

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FE00, 0x00FF00, 0x01FF00, 0x01FF80, 0x01FF80, 0x03FF80, 0x03EFC0, 0x03E7C0, 0x07E7C0, 0x07C7E0, 0x0FC3E0, 0x0FC3F0, 0x0FFFF0, 0x1FFFF0, 0x1FFFF8, 0x1FFFF8, 0x3F00F8, 0x3F00FC, 0x7E00FC, 0x7E007E},

//'B

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1FFF80, 0x1FFFC0, 0x1FFFE0, 0x1FFFE0, 0x1F87F0, 0x1F83F0, 0x1F83F0, 0x1F87E0, 0x1FFFE0, 0x1FFFC0, 0x1FFFE0, 0x1FFFF0, 0x1F83F0, 0x1F81F8, 0x1F81F8, 0x1F83F8, 0x1FFFF0, 0x1FFFF0, 0x1FFFE0, 0x1FFF80},

//'C

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007FC0, 0x01FFF0, 0x07FFF0, 0x0FFFF0, 0x0FF0F0, 0x1FC030, 0x1F8000, 0x3F8000, 0x3F0000, 0x3F0000, 0x3F0000, 0x3F0000, 0x3F8000, 0x1F8000, 0x1FC030, 0x0FF0F0, 0x0FFFF0, 0x07FFF0, 0x01FFF0, 0x007FC0},

//'D

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3FFE00, 0x3FFFC0, 0x3FFFE0, 0x3FFFF0, 0x3F0FF8, 0x3F03F8, 0x3F01FC, 0x3F00FC, 0x3F00FC, 0x3F00FC, 0x3F00FC, 0x3F00FC, 0x3F00FC, 0x3F01FC, 0x3F03F8, 0x3F0FF8, 0x3FFFF0, 0x3FFFE0, 0x3FFFC0, 0x3FFE00},

//'E

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0},

//'F

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FFFE0, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000},

//'G

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FFC0, 0x03FFF0, 0x0FFFF0, 0x1FFFF0, 0x1FE0F0, 0x3F8010, 0x3F0000, 0x7F0000, 0x7E0000, 0x7E0FF8, 0x7E0FF8, 0x7E0FF8, 0x7F0FF8, 0x3F00F8, 0x3F80F8, 0x1FE1F8, 0x1FFFF8, 0x0FFFF8, 0x03FFF0, 0x00FFC0},

//'H

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3FFFF8, 0x3FFFF8, 0x3FFFF8, 0x3FFFF8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8, 0x3F01F8},

//'I

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00},

//'J

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x03FC00, 0x0FF800},

//'K

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3F01F8, 0x3F03F0, 0x3F0FE0, 0x3F1FC0, 0x3F3F80, 0x3F7F00, 0x3FFE00, 0x3FFC00, 0x3FF800, 0x3FF000, 0x3FF800, 0x3FFC00, 0x3FFE00, 0x3F7F00, 0x3F3F80, 0x3F1FC0, 0x3F0FE0, 0x3F07F0, 0x3F03F8, 0x3F01FC},

//'L

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07E000, 0x07FFF0, 0x07FFF0, 0x07FFF0, 0x07FFF0},

//'M

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0xFF00FF, 0xFF00FF, 0xFF81FF, 0xFF81FF, 0xFFC3FF, 0xFFC3FF, 0xFFC7FF, 0xFFE7FF, 0xFDE7BF, 0xFDFFBF, 0xFCFF3F, 0xFCFF3F, 0xFCFE3F, 0xFC7E3F, 0xFC7E3F, 0xFC3C3F, 0xFC003F, 0xFC003F, 0xFC003F, 0xFC003F},

//'N

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3F80F8, 0x3FC0F8, 0x3FC0F8, 0x3FE0F8, 0x3FE0F8, 0x3FF0F8, 0x3FF0F8, 0x3FF8F8, 0x3FF8F8, 0x3F7CF8, 0x3F3CF8, 0x3F3EF8, 0x3F1FF8, 0x3F1FF8, 0x3F0FF8, 0x3F0FF8, 0x3F07F8, 0x3F07F8, 0x3F03F8, 0x3F03F8},

//'O

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF80, 0x07FFC0, 0x0FFFF0, 0x1FFFF8, 0x3FC7F8, 0x3F81FC, 0x3F00FC, 0x7E00FC, 0x7E007E, 0x7E007E, 0x7E007E, 0x7E007E, 0x7E00FC, 0x3F00FC, 0x3F81FC, 0x3FC7F8, 0x1FFFF8, 0x0FFFF0, 0x07FFC0, 0x01FF80},

//'P

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1FFF80, 0x1FFFC0, 0x1FFFE0, 0x1FFFF0, 0x1F87F0, 0x1F83F8, 0x1F81F8, 0x1F83F8, 0x1F87F0, 0x1FFFF0, 0x1FFFE0, 0x1FFFC0, 0x1FFF80, 0x1F8000, 0x1F8000, 0x1F8000, 0x1F8000, 0x1F8000, 0x1F8000, 0x1F8000},

//'Q

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF80, 0x07FFE0, 0x0FFFF0, 0x1FFFF8, 0x3FC7F8, 0x3F81FC, 0x3F00FC, 0x7E00FC, 0x7E007E, 0x7E007E, 0x7E007E, 0x7E007E, 0x7E00FC, 0x3F00FC, 0x3F81FC, 0x3FC7F8, 0x1FFFF8, 0x0FFFF0, 0x07FFC0, 0x01FF80, 0x0007C0, 0x0007E0},

//'R

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1FFF00, 0x1FFFC0, 0x1FFFE0, 0x1FFFE0, 0x1F87F0, 0x1F83F0, 0x1F83F0, 0x1F83E0, 0x1F87E0, 0x1FFFC0, 0x1FFF80, 0x1FFF80, 0x1FFFC0, 0x1F8FE0, 0x1F87E0, 0x1F83F0, 0x1F83F0, 0x1F81F8, 0x1F81F8, 0x1F80FC},

//'S

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FFC0, 0x03FFE0, 0x07FFE0, 0x0FFFE0, 0x0FE3E0, 0x0F8060, 0x0F8000, 0x0FC000, 0x0FFE00, 0x07FFC0, 0x03FFE0, 0x00FFF0, 0x0007F0, 0x0001F0, 0x0C01F0, 0x0F87F0, 0x0FFFF0, 0x0FFFE0, 0x0FFFC0, 0x03FF80},

//'T

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3FFFF8, 0x3FFFF8, 0x3FFFF8, 0x3FFFF8, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00},

//'U

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F01F0, 0x3F03F0, 0x1F83F0, 0x1FC7F0, 0x1FFFE0, 0x0FFFE0, 0x07FFC0, 0x01FF00},

//'V

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x7E007E, 0x7E00FC, 0x3F00FC, 0x3F00F8, 0x1F01F8, 0x1F81F8, 0x1F81F0, 0x0F83F0, 0x0FC3F0, 0x0FC7E0, 0x07E7E0, 0x07E7C0, 0x03EFC0, 0x03FFC0, 0x03FF80, 0x01FF80, 0x01FF80, 0x01FF00, 0x00FF00, 0x00FE00},

//'W

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0xC07C07, 0xE0FE0F, 0xE0FE0F, 0xE0FE0F, 0xE0FE0F, 0xE1FF1F, 0xF1EF1F, 0xF1EF1F, 0xF1EF1F, 0xF3EFBF, 0xFBC7BF, 0xFBC7BF, 0xFBC7BE, 0xFFC7FE, 0xFF83FE, 0xFF83FE, 0x7F83FC, 0x7F83FC, 0x7F03FC, 0x7F01FC},

//'X

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x3F00FC, 0x1F81F8, 0x1F83F0, 0x0FC3F0, 0x07E7E0, 0x07EFC0, 0x03FFC0, 0x01FF80, 0x01FF00, 0x00FE00, 0x00FF00, 0x01FF00, 0x03FF80, 0x03FFC0, 0x07EFE0, 0x0FC7E0, 0x0FC3F0, 0x1F81F8, 0x3F01F8, 0x3F00FC},

//'Y

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x7E01FC, 0x7F01F8, 0x3F03F0, 0x1F87E0, 0x1FC7E0, 0x0FCFC0, 0x07FF80, 0x07FF80, 0x03FF00, 0x01FE00, 0x01FE00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00},

//'Z

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x0007F0, 0x0007E0, 0x000FC0, 0x001F80, 0x003F00, 0x007F00, 0x00FE00, 0x00FC00, 0x01F800, 0x03F000, 0x07E000, 0x0FE000, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8, 0x1FFFF8},

//'[

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x01FF00, 0x01FF00, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01F000, 0x01FF00, 0x01FF00},

//'\

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x038000, 0x03C000, 0x03C000, 0x01C000, 0x01E000, 0x01E000, 0x00E000, 0x00F000, 0x00F000, 0x007000, 0x007800, 0x007800, 0x003800, 0x003C00, 0x003C00, 0x001C00, 0x001C00, 0x001E00, 0x001E00, 0x000E00, 0x000F00, 0x000F00},

//']

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x01FF00, 0x01FF00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x003F00, 0x01FF00, 0x01FF00},

//'^

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007C00, 0x00FE00, 0x01FF00, 0x01FF80, 0x03EFC0, 0x07C3E0, 0x0F01F0, 0x1E00F0, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'\_

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'`

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x03E000, 0x01F000, 0x00F000, 0x007800, 0x003C00, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000},

//'a

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07FF00, 0x0FFF80, 0x0FFFC0, 0x0E0FE0, 0x0007E0, 0x03FFE0, 0x0FFFE0, 0x0FFFE0, 0x1FC7E0, 0x1F87E0, 0x1F87E0, 0x1F8FE0, 0x0FFFE0, 0x03FFC0, 0x01FF80},

//'b

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FDF80, 0x0FFFE0, 0x0FFFF0, 0x0FE3F0, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FE3F0, 0x0FFFF0, 0x0FFFE0, 0x0FDF80, 0x000000},

//'c

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007F80, 0x01FFC0, 0x03FFC0, 0x07F1C0, 0x0FE000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FE000, 0x07F0C0, 0x03FFC0, 0x03FFC0, 0x01FF80},

//'d

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0003F0, 0x0003F0, 0x0003F0, 0x0003F0, 0x0003F0, 0x0003F0, 0x01FBF0, 0x07FFF0, 0x07FFF0, 0x0FC7F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x0FC7F0, 0x0FFFF0, 0x07FFF0, 0x01FBF0, 0x000000},

//'e

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FE00, 0x03FF80, 0x07FFC0, 0x0FC7E0, 0x1F83E0, 0x1F83F0, 0x1FFFF0, 0x1FFFF0, 0x1FFFF0, 0x1F8000, 0x1F8000, 0x0FE1E0, 0x07FFE0, 0x03FFC0, 0x01FF80},

//'f

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x003FC0, 0x007FC0, 0x00FFC0, 0x00FC00, 0x00F800, 0x00F800, 0x07FF80, 0x07FF80, 0x07FF80, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x000000},

//'g

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FBF0, 0x07FFF0, 0x07FFF0, 0x0FC7F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x0FC7F0, 0x07FFF0, 0x07FFF0, 0x01FBF0, 0x0003F0, 0x0003F0, 0x0003F0, 0x001FF0, 0x00FFF0, 0x00FFB0, 0x03FF80},

//'h

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FCF80, 0x0FFFE0, 0x0FFFE0, 0x0FE7F0, 0x0FC3F0, 0x0FC3F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x000000},

//'i

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x000000, 0x000000, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x000000},

//'j

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00FC00, 0x00FC00, 0x000000, 0x000000, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00FC00, 0x00F800, 0x00F800, 0x01F800, 0x07F800, 0x07F000, 0x0FE000},

//'k

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC3F0, 0x0FC7E0, 0x0FCFC0, 0x0FDF80, 0x0FFF00, 0x0FFE00, 0x0FFC00, 0x0FFC00, 0x0FFE00, 0x0FFF00, 0x0FDF80, 0x0FCFC0, 0x0FCFE0, 0x0FC7F0, 0x0FC3F8, 0x000000},

//'l

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x000000},

//'m

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0xFBF0FC, 0xFFFBFE, 0xFFFFFF, 0xFCFF3F, 0xF87E1F, 0xF87E1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F, 0xF87C1F},

//'n

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FCF80, 0x0FFFE0, 0x0FFFE0, 0x0FE7F0, 0x0FC3F0, 0x0FC3F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0, 0x0FC1F0},

//'o

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x03FFC0, 0x07FFE0, 0x0FC7E0, 0x1F83F0, 0x1F83F0, 0x1F81F0, 0x1F81F0, 0x1F81F0, 0x1F83F0, 0x1F83F0, 0x0FC7E0, 0x07FFE0, 0x07FFC0, 0x03FF80},

//'p

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FDF80, 0x0FFFE0, 0x0FFFF0, 0x0FE3F0, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FC1F8, 0x0FE3F0, 0x0FFFF0, 0x0FFFE0, 0x0FDF80, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000, 0x0FC000},

//'q

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FBF0, 0x07FFF0, 0x0FFFF0, 0x0FC7F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x1F83F0, 0x0FC7F0, 0x0FFFF0, 0x07FFF0, 0x01FBF0, 0x0003F0, 0x0003F0, 0x0003F0, 0x0003F0, 0x0003F0},

//'r

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FBE0, 0x01FFE0, 0x01FFE0, 0x01FFE0, 0x01FE60, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800, 0x01F800},

//'s

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x01FF00, 0x07FFC0, 0x07FFC0, 0x0FC3C0, 0x0F8000, 0x0FF800, 0x07FF00, 0x07FFC0, 0x03FFC0, 0x003FE0, 0x0007E0, 0x0F07C0, 0x0FFFC0},

//'t

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x07FFC0, 0x07FFC0, 0x07FFC0, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00F800, 0x00FC00, 0x00FFC0, 0x00FFC0, 0x003FC0},

//'u

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0F83F0, 0x0FC3F0, 0x0FE7F0, 0x07FFF0, 0x07FFF0, 0x07FFF0, 0x07FFF0},

//'v

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1F81F8, 0x0F81F8, 0x0FC1F0, 0x07C3F0, 0x07C3E0, 0x07E3E0, 0x03E7E0, 0x03E7C0, 0x01F7C0, 0x01FF80, 0x01FF80, 0x00FF00, 0x00FF00, 0x00FF00, 0x00FF00},

//'w

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0xF87C1F, 0xF87E3F, 0xFC7E3E, 0x7C7E3E, 0x7CFE3E, 0x7CFF7E, 0x7EFF7C, 0x3EFF7C, 0x3FE77C, 0x3FE7FC, 0x1FE7F8, 0x1FE7F8, 0x1FC3F8, 0x1FC3F8, 0x1FC3F8},

//'x

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x0FC1F0, 0x07C3F0, 0x07E7E0, 0x03FFC0, 0x01FF80, 0x00FF80, 0x00FF00, 0x007E00, 0x00FF00, 0x01FF80, 0x01FFC0, 0x03E7C0, 0x07E3E0, 0x0FC3E0, 0x1F83F0, 0x3E01F8, 0x3E01F8},

//'y

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x1F81F8, 0x0F81F0, 0x0FC1F0, 0x07C3F0, 0x07C3E0, 0x07E3E0, 0x03E7E0, 0x03F7C0, 0x01F7C0, 0x01FF80, 0x00FF80, 0x00FF80, 0x007F00, 0x007F00, 0x007E00, 0x003E00, 0x003E00, 0x007C00, 0x00F000, 0x00F000, 0x1FF000, 0x1FF000},

//'z

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07FFE0, 0x07FFE0, 0x07FFE0, 0x0007E0, 0x000FC0, 0x001F80, 0x003F00, 0x007E00, 0x00FE00, 0x01FC00, 0x03F800, 0x07F000, 0x07FFE0, 0x07FFE0, 0x07FFE0, 0x07FFE0},

//'{

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000FE0, 0x001FE0, 0x003FE0, 0x003F00, 0x003E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x00FC00, 0x07FC00, 0x07F800, 0x07FC00, 0x00FC00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x007E00, 0x003F00, 0x003FE0},

//'|

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00, 0x003C00},

//'}

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07F000, 0x07F800, 0x07FC00, 0x00FC00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x007E00, 0x007E00, 0x003F00, 0x003FE0, 0x001FE0, 0x003FE0, 0x003F00, 0x007E00, 0x007E00, 0x007C00, 0x007C00, 0x007C00, 0x007C00, 0x00FC00, 0x07FC00},

//'~

{0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x000000, 0x07E018, 0x1FFC78, 0x1FFFF8, 0x1C7FF0, 0x100FC0, 0x000000, 0x000000}

};

unsigned long long int temp;

temp=c-32;

unsigned long long int \*char\_data=ascii\_font[temp];

unsigned long long int i,j;

unsigned long long int record=1;

unsigned long long int recording=CHAR\_WIDTH;

if(status==0)

{

for(i=0;i<CHAR\_HEIGHT;i++)

{

for(j=0;j<CHAR\_WIDTH;j++)

{

if(char\_data[i]&(record<<(recording-record-j)))

{

point\_drawing(x+j,y+i,color,page);

}

}

}

}

else if(status==1)

{

for(i=0;i<CHAR\_HEIGHT;i++)

{

for(j=0;j<CHAR\_WIDTH;j++)

{

if(char\_data[i]&(record<<(recording-record-j)))

{

if(y+i<=575)

{

point\_drawing(x+j,y+i,color,page);

}

}

}

}

}

}

//void English\_text\_drawing24(int x,int y,char c,int color,int page)

//{

//}

void English\_text\_drawing(int x,int y,const char \*text,int color,int page,int status)

{

while(\*text!='\0')

{

English\_drawing(x,y,\*text,color,page,status);

x+=CHAR\_WIDTH;

text++;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 函数名称：interrupt

\*\* 函数目的：通过对互信息值的确定，实现断句。

\*\* 时间复杂度：OH(字数)

\*\* 空间复杂度：OH(字数)+6766

\*/

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<page.h>

#include<xiezi.h>

#include<color.h>

#define MI\_VALUE 0.0000000000001

#define ROW\_LENGTH 87921

#define EACH\_LENGTH 13

#define INPUT\_SIZE 10000

#define GB2312\_NUMBER 6763

double mi\_value(int i,int j)

{

//打开文件

FILE \*fp;

if((fp=fopen("C:\\CODE\\data\\tn","rt"))==NULL)

{

printf("GB2312互信息值数据库无法打开！任意键退出");

getchar();

return -1;

}

//找到位置，读取

long long int calculate;

calculate=i\*ROW\_LENGTH+j\*EACH\_LENGTH+1;

//calculate=87919;

fseek(fp,calculate,0);

//记录字符串，然后变成小数

char record[12];

for(i=0;i<12;i++)

{

record[i]=fgetc(fp);

}

record[12]='\0';

double result;

result=atof(record);

fclose(fp);

return result;

}

int fenci(char\* input,char\* eventual)

{

char arr[GB2312\_NUMBER][2];

FILE \*fp;

int i;

char zi5[5]={0};

if((fp=fopen("C:\\CODE\\data\\GB2312","r"))==NULL)

{

printf("GB2312数据库无法打开！任意键退出\n");

getchar();

return -1;

}

for(i=0;i<GB2312\_NUMBER;i++)

{

for(int j=0;j<2;j++)

{

arr[i][j]=fgetc(fp);

}

}

//int \*count=(int\*)malloc(INPUT\_SIZE/2);

int count[INPUT\_SIZE/2];

int counting=0;

int flag=0;

char temp[5];

for(i=0;i<strlen(input);i+=2)

{

if(input[i]>0)//就是不是汉字的意思

{

count[counting++]=0;//0是标志句子分开

//i--;

}

else

{

for(int j=0;j<GB2312\_NUMBER;j++)

{

if(input[i]==arr[j][0])

{

if(input[i+1]==arr[j][1])

{

count[counting++]=j;//最后储存的从0开始储存

}

}

}

}

flag++;

//每执行一遍函数，count多记一个，flag加1,flag的数值等于count数组元素的个数

}

for(i=0;i<counting;i++)

{

printf("%d",count[i]);

printf("\n");

}

for(i=0;i<strlen(input);i+=2)

{

if(input[i]>0)//就是不是汉字的意思

{

count[counting++]=0;//0是标志句子分开

//i--;

}

else

{

for(int j=0;j<GB2312\_NUMBER;j++)

{

if(input[i]==arr[j][0])

{

if(input[i+1]==arr[j][1])

{

count[counting++]=j;//最后储存的从0开始储存

}

}

}

}

flag++;

//每执行一遍函数，count多记一个，flag加1,flag的数值等于count数组元素的个数

}

int record=-2;

counting=0;

unsigned int char\_code;

//下面开始断句的内容

for(i=0;i<flag+1;i++)

{

//标点全断，但是未考虑英文和数字

if(input[record]>0)

{

record++;

//printf("||,||");//此处内容还可以根据需求改

strcat(eventual,"||");

// ||,||

}

//第一个字是汉字，第二个是标点，这样就不在后面断了

else if(input[record]<0&&input[record+2]>=0)

{

record+=2;

//printf("%c%c",input[record],input[record+1]);

temp[0]=input[record];

temp[1]=input[record+1];

temp[2]='\0';

strcat(eventual,temp);

}

else

{

//两个字都是汉字

record+=2;

if(mi\_value(count[counting],count[counting+1])==-1)

{

printf("再按一次，彻底退出");

getchar();

return 1;

}

if(mi\_value(count[counting],count[counting+1])>=MI\_VALUE)

{

//本应连在一起的字

zi5[0]=input[record];

zi5[1]=input[record+1];

if(strcmp(zi5,"是")==0||

strcmp(zi5,"有")==0||

strcmp(zi5,"谁")==0||

strcmp(zi5,"啥")==0)

//常见单音节词，容易连东西，但是应该分开

//不输入我，你等人称代词，是因为有我们，你们这种集合人称代词

{

temp[0]=input[record];

temp[1]=input[record+1];

temp[2]='|';

temp[3]='|';

temp[4]='\0';

strcat(eventual,temp);

}

else

{

zi5[0]=input[record+2];

zi5[1]=input[record+3];

if(strcmp(zi5,"是")==0||

strcmp(zi5,"有")==0||

strcmp(zi5,"谁")==0||

strcmp(zi5,"啥")==0)

{

temp[0]=input[record];

temp[1]=input[record+1];

temp[2]='|';

temp[3]='|';

temp[4]='\0';

strcat(eventual,temp);

}

else

{

temp[0]=input[record];

temp[1]=input[record+1];

temp[2]='\0';

strcat(eventual,temp);

}

}

}

else

{

//本不应连在一起的字

//printf("%c%c||",input[record],input[record+1]);

temp[0]=input[record];

temp[1]=input[record+1];

temp[2]='|';

temp[3]='|';

temp[4]='\0';

strcat(eventual,temp);

}

}

counting++;

}

fclose(fp);

for(i=strlen(eventual)-1;i>=0;i--)

{

if(eventual[i]>=0)

{

eventual[i]='\0';

}

else

{

break;

}

}

printf("%s\n",eventual);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：find

\*\*文件目的：获得lasthao相对应的逻辑词语

\*\*时间复杂度：k1\*树的高度+k2\*词语个数

\*\*空间复杂度：常数

\*/

#include<lasthao.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<time.h>

#include<find.h>

#include<xiezi.h>

#include<page.h>

#include<dos.h>

//函数名称：find\_logic

//函数目的：实现单个句子的文献查找

char\* find\_logic(ZHUWEIBIN arr,char\* eventual,char\* result)

{

int i;

FILE\* fp;

char temp;

char rchar;

int rrr;

int counting=0;

int cishu=0;

static char shuzu[400]={0};

for(i=0;i<550;i++)

{

shuzu[i]='\0';

}

char weiyucici[50];

char binyucici[50];

memset(weiyucici,0,50);

memset(binyucici,0,50);

char transit[200]={0};

char str[100]={0};

int branch=panduanshenmeci(arr);

strcpy(str,file\_zhu(arr));

char filechar;

int suiji;

int filenum=0;

int zanghua=0;

int sss=0;

//char hahaha[100]={0};

//itoa(arr.wei[1],hahaha,10);

//strcpy(shuzu,str);

//return shuzu;

fp=fopen(str,"rb");

//delay(10);

if(fp==NULL)

{

if(strstr(eventual,"我靠")!=NULL)

{

strcpy(shuzu,"莫说脏话！");

zanghua=26;

sss=1;

}

else if(strstr(eventual,"傻逼")!=NULL)

{

strcpy(shuzu,"莫说脏话！");

zanghua=26;

}

else if(strstr(eventual,"我草")!=NULL)

{

strcpy(shuzu,"莫说脏话！");

zanghua=26;

sss=1;

}

else if(strstr(eventual,"妈的")!=NULL)

{

strcat(shuzu,"莫说脏话！");

zanghua=26;

}

if(strstr(eventual,"请你")!=NULL)

{

if(sss==1)

{

goto shelter;

}

replace(eventual,"请你","这就");

strcpy(shuzu,eventual);

replace(eventual,"这就","请你");

return shuzu;

}

if(strstr(eventual,"我")!=NULL)

{

if(sss==1)

{

goto shelter;

}

replace(eventual,"我","你");

strcpy(shuzu,eventual);

replace(eventual,"你","我");

return shuzu;

}

if(strstr(eventual,"你")!=NULL)

{

if(sss==1)

{

goto shelter;

}

replace(eventual,"你","我");

strcpy(shuzu,eventual);

replace(eventual,"我","你");

return shuzu;

}

shelter:

srand(time(0));

suiji=rand()%5;

if(suiji==0)

{

strcpy(shuzu,"抱歉我没听懂你的意思");

}

else if(suiji==1)

{

strcpy(shuzu,"抱一丝没听懂");

}

else if(suiji==2)

{

strcpy(shuzu,"你说的啥？木有搞懂");

}

else if(suiji==3)

{

strcpy(shuzu,"单纯的小AI我听不懂聪明的你的话语");

}

else if(suiji==4)

{

strcpy(shuzu,"啥？？？没听懂");

}

return shuzu;

}

//找疑问语气的内容

if(branch==0)

{

if(strstr(eventual,"我靠")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"傻逼")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"我草")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strcpy(eventual,"妈的")!=NULL)

{

strcat(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

//strcpy(shuzu,"branch=0");

//return shuzu;

if(strstr(eventual,"是")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

}

}

else if(filechar=='$'&&filenum==1)

{

fclose(fp);

goto x;

}

if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

if(strstr(eventual,"什么")!=NULL)

{

//memset(shuzu,0,500);

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

//exit(0);

}

}

else if(filechar=='9'&&filenum==1)

{

fclose(fp);

goto x1;

break;

}

else if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"怎")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"如何")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"咋")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

x:

shuzu[strlen(shuzu)-1]='\0';

return shuzu;

}

goto gogogo;

//这就是只有主语

}

else if(branch==1)

{

if(strstr(eventual,"我靠")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"傻逼")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"我草")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"妈的")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

//strcpy(shuzu,"branch=1");

//return shuzu;

if(strstr(eventual,"是")!=NULL)

{

//memset(shuzu,0,500);

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

//exit(0);

}

}

else if(filechar=='9'&&filenum==1)

{

fclose(fp);

goto x1;

break;

}

else if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

if(strstr(eventual,"什么")!=NULL)

{

//memset(shuzu,0,500);

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

//exit(0);

}

}

else if(filechar=='9'&&filenum==1)

{

fclose(fp);

goto x1;

break;

}

else if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"怎")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"如何")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

else if(strstr(eventual,"咋")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

x1:

shuzu[strlen(shuzu)-1]='\0';

//fclose(fp);

return shuzu;

}

goto gogogo;

}

else

{

gogogo:

if(strstr(eventual,"我靠")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"傻逼")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"我草")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

else if(strstr(eventual,"妈的")!=NULL)

{

strcpy(shuzu,"首先请文明的你不要说脏话。");

zanghua=26;

}

if(strstr(eventual,"是")!=NULL)

{

//memset(shuzu,0,500);

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

//exit(0);

}

}

else if(filechar=='9'&&filenum==1)

{

fclose(fp);

//shuzu[strlen(shuzu)-1]='\0';

goto x1;

break;

}

else if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

if(strstr(eventual,"什么")!=NULL)

{

//memset(shuzu,0,500);

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0)

{

filenum=1;

//exit(0);

}

}

else if(filechar=='9'&&filenum==1)

{

fclose(fp);

goto x1;

break;

}

else if(filenum==1)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

}

if(strstr(eventual,"怎")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x2;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

x2:

shuzu[strlen(shuzu)-1]='\0';

//fclose(fp);

return shuzu;

}

if(strstr(eventual,"如何")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x3;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

x3:

shuzu[strlen(shuzu)-1]='\0';

//fclose(fp);

return shuzu;

}

if(strstr(eventual,"咋")!=NULL)

{

while(1)

{

filechar=fgetc(fp);

if(filechar=='5')

{

if(filenum==0||filenum==1)

{

filenum++;

}

}

else if(filechar=='$'&&filenum==2)

{

fclose(fp);

goto x4;

}

else if(filenum==2)

{

shuzu[strlen(shuzu)]=filechar;

//shuzu[strlen(shuzu)]='\0';

}

}

x4:

shuzu[strlen(shuzu)-1]='\0';

//fclose(fp);

return shuzu;

}

memset(shuzu,0,550);

//所谓有宾语，无非是观察词与词的联系

if(arr.bin[2]==1)

{

while(1)

{

temp=fgetc(fp);

if(strstr(str,"$")==NULL)

{

if(temp=='$')

{

strcat(str,"$");

}

continue;

}

if(strstr(str,"$")!=NULL&&strstr(str,"$9")==NULL)

{

if(temp=='9')

{

strcat(str,"9");

}

else

{

str[0]='\0';

}

continue;

}

if(strstr(str,"$9")!=NULL&&strstr(str,"$98")==NULL)

{

if(temp=='8')

{

strcat(str,"8");

}

else

{

str[0]='\0';

str[1]='\0';

}

continue;

}

if(strstr(str,"$98")!=NULL&&strstr(str,"$985")==NULL)

{

if(temp=='5')

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

break;

}

else

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

continue;

}

}

if(feof(fp))

{

break;

}

}

while(1)

{

temp=fgetc(fp);

if(temp=='$')

{

counting=0;

break;

}

else

{

result[counting++]=temp;

}

if(feof(fp))

{

break;

}

}

strcpy(shuzu,result);

fclose(fp);

return shuzu;

}

//怎么做部分

else if(arr.bin[2]==3)

{

while(1)

{

temp=fgetc(fp);

if(strstr(str,"$")==NULL)

{

if(temp=='$')

{

strcat(str,"$");

}

continue;

}

if(strstr(str,"$")!=NULL&&strstr(str,"$9")==NULL)

{

if(temp=='9')

{

strcat(str,"9");

}

else

{

str[0]='\0';

}

continue;

}

if(strstr(str,"$9")!=NULL&&strstr(str,"$98")==NULL)

{

if(temp=='8')

{

strcat(str,"8");

}

else

{

str[0]='\0';

str[1]='\0';

}

continue;

}

if(strstr(str,"$98")!=NULL&&strstr(str,"$985")==NULL)

{

if(temp=='5')

{

if(cishu==0)

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

cishu++;

}

else

{

cishu=0;

str[0]='\0';

str[1]='\0';

str[2]='\0';

break;

}

}

else

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

continue;

}

}

if(feof(fp))

{

break;

}

}

while(1)

{

temp=fgetc(fp);

if(temp=='$')

{

counting=0;

break;

}

else

{

result[counting++]=temp;

}

if(feof(fp))

{

break;

}

}

strcpy(shuzu,result);

fclose(fp);

return shuzu;

}

//其他部分

else

{

while(1)

{

temp=fgetc(fp);

if(feof(fp))

{

goto shelter;

break;

}

if(strstr(str,"$")==NULL)

{

if(temp=='$')

{

strcat(str,"$");

}

//goto shelter;

continue;

}

if(strstr(str,"$")!=NULL&&strstr(str,"$9")==NULL)

{

if(temp=='9')

{

strcat(str,"9");

}

else

{

str[0]='\0';

}

//goto shelter;

continue;

}

if(strstr(str,"$9")!=NULL&&strstr(str,"$98")==NULL)

{

if(temp=='8')

{

strcat(str,"8");

}

else

{

str[0]='\0';

str[1]='\0';

}

//goto shelter;

continue;

}

if(strstr(str,"$98")!=NULL&&strstr(str,"$985")==NULL)

{

if(temp=='5')

{

if(cishu==0||cishu==1)

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

cishu++;

//goto shelter;

continue;

}

else

{

cishu=0;

str[0]='\0';

str[1]='\0';

str[2]='\0';

//goto shelter;

break;

}

}

else

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

//goto shelter;

continue;

}

}

}

strcpy(weiyucici,weiyuci(arr));

FILE \*fpwei;

fpwei=fopen(weiyucici,"rt+");

if(fpwei==NULL)

{

goto shelter;

}

char weiyurecord[50]={0};

int hhh;

char hhhwei;

for(hhh=0;;hhh++)

{

hhhwei=fgetc(fpwei);

if(hhhwei!='$')

{

weiyurecord[hhh]=hhhwei;

}

else

{

break;

}

}

strcpy(weiyucici,weiyurecord);

//上面成功把指针引到了最后的部分

while(1)

{

temp=fgetc(fp);

//此处须改变一下，后面的那个改成编码

if(strstr(str,"-")==NULL)

{

if(temp=='-')

{

strcat(str,"-");

}

continue;

}

if(strstr(str,"-=")==NULL&&strstr(str,"-")!=NULL)

{

if(temp=='=')

{

strcat(str,"=");

}

else

{

str[0]='\0';

}

continue;

}

if(strstr(str,"-=")!=NULL&&strstr(str,"-=+")==NULL)

{

if(temp=='+')

{

strcat(str,"+");

continue;

}

else

{

result[counting++]=temp;

//result[counting++]='\0';

continue;

}

}

if(strstr(str,"-=+")!=NULL&&strstr(str,"-=+=")==NULL)

{

if(temp=='='&&!feof(fp))

{

result[counting++]='\0';

if(strstr(weiyucici,result)!=NULL)

{

rrr=0;

while(1)

{

rchar=fgetc(fp);

if(rchar!='-')

{

shuzu[rrr++]=rchar;

}

else

{

shuzu[rrr]='\0';

break;

}

}

fclose(fp);

fclose(fpwei);

return shuzu;

}

//result[counting++]=temp;

//continue;

}

else if(feof(fp))//到达文件末尾

{

//此为异常情况，进行标记，因为宾语为空

int j;

shuzu[0]=1;

shuzu[1]=1;

shuzu[2]=4;

shuzu[3]=5;

shuzu[4]=1;

shuzu[5]=4;

goto shelter;

break;

}

else

{

str[0]='\0';

str[1]='\0';

str[2]='\0';

memset(result,0,sizeof(result));

continue;

}

}/\*

if(strstr(str,"-=+=")!=NULL&&strstr(str,"-=+=-")==NULL)

{

if(temp=='-'&&!feof(fp))

{

strcat(str,"-");

continue;

}

else if(feof(fp))

{

if(strcmp(weiyucici,result)==0)

{

strcpy(shuzu,result);

}

else

{

//没找到词的话......

goto shelter;

}

break;

}

else

{

result[counting++]=temp;

continue;

}

}

if(strstr(str,"-=+=-")!=NULL)

{

if(temp=='=')

{

int j;

for(j=2;j<6;j++)

{

str[j]='\0';

}

//注意下一次还得用str[0]和str[1]呢！

if(strcmp(result,weiyucici)==0)

{

strcpy(shuzu,result);

memset(result,0,sizeof(result));

break;

}

else

{

counting=0;

continue;

}

}

else

{

str[4]='\0';

result[counting++]='-';

result[counting++]=temp;

}

}\*/

}

}

//以上的基本假设是，里面不会有-=,+=符号的真正内容

fclose(fp);

return shuzu;

}

fclose(fp);

return shuzu;

}

//函数名称：file\_zhu

//函数目的：得出主语的文件名

char\* file\_zhu(ZHUWEIBIN arr)

{

char str1[100];

char str2[100];

char str3[100];

char str4[100];

itoa(arr.zhu[7],str2,10);

itoa(arr.zhu[8],str3,10);

itoa(arr.zhu[9],str4,10);

strcpy(str1,"C:\\CODE\\inf\\");

strcat(str1,str2);

strcat(str1,"\\");

strcat(str1,str2);

strcat(str1,"-");

strcat(str1,str3);

strcat(str1,"\\");

strcat(str1,str2);

strcat(str1,"-");

strcat(str1,str3);

strcat(str1,"-");

strcat(str1,str4);

strcat(str1,".txt");

return str1;

}

//函数名称：file\_wei

//函数目的：得出谓语的文件名

char\* file\_wei(ZHUWEIBIN arr)

{

char str11[100];

char str2[100];

char str3[100];

char str4[100];

itoa(arr.wei[6],str2,10);

itoa(arr.wei[7],str3,10);

itoa(arr.wei[8],str4,10);

strcpy(str11,"C:\\CODE\\inf\\");

strcat(str11,str2);

strcat(str11,"\\");

strcat(str11,str2);

strcat(str11,"-");

strcat(str11,str3);

strcat(str11,"\\");

strcat(str11,str2);

strcat(str11,"-");

strcat(str11,str3);

strcat(str11,"-");

strcat(str11,str4);

strcat(str11,".txt");

return str11;

}

//函数名称：file\_bin

//函数目的：得出宾语的文件名

char\* file\_bin(ZHUWEIBIN arr)

{

char str111[100];

char str2[100];

char str3[100];

char str4[100];

itoa(arr.zhu[6],str2,10);

itoa(arr.zhu[7],str3,10);

itoa(arr.zhu[8],str4,10);

strcpy(str111,"C:\\CODE\\inf\\");

strcat(str111,str2);

strcat(str111,"\\");

strcat(str111,str2);

strcat(str111,"-");

strcat(str111,str3);

strcat(str111,"\\");

strcat(str111,str2);

strcat(str111,"-");

strcat(str111,str3);

strcat(str111,"-");

strcat(str111,str4);

strcat(str111,".txt");

return str111;

}

//函数名称：weiyuci

//函数目的：得出谓语词标号对应的char\*字符串

char\* weiyuci(ZHUWEIBIN arr)

{

static char temp1[50];

memset(temp1,0,50);

char shuzu1[50];

char shuzu2[50];

char shuzu3[50];

itoa(arr.wei[7],shuzu1,10);

itoa(arr.wei[8],shuzu2,10);

itoa(arr.wei[9],shuzu3,10);

strcpy(temp1,"C:\\CODE\\inf\\");

strcat(temp1,shuzu1);

strcat(temp1,"\\");

strcat(temp1,shuzu1);

strcat(temp1,"-");

strcat(temp1,shuzu2);

strcat(temp1,"\\");

strcat(temp1,shuzu1);

strcat(temp1,"-");

strcat(temp1,shuzu2);

strcat(temp1,"-");

strcat(temp1,shuzu3);

strcat(temp1,".txt");

return temp1;

}

//函数名称：binyuci

//函数目的：得出宾语词标号对应的char\*字符串

char\* binyuci(ZHUWEIBIN arr)

{

static char temp1[50];

memset(temp1,0,50);

char shuzu1[50];

char shuzu2[50];

char shuzu3[50];

itoa(arr.wei[6],shuzu1,10);

itoa(arr.wei[7],shuzu2,10);

itoa(arr.wei[8],shuzu3,10);

strcpy(temp1,"C:\\CODE\\inf\\");

strcat(temp1,shuzu1);

strcat(temp1,"\\");

strcat(temp1,shuzu1);

strcat(temp1,"-");

strcat(temp1,shuzu2);

strcat(temp1,"\\");

strcat(temp1,shuzu1);

strcat(temp1,"-");

strcat(temp1,shuzu2);

strcat(temp1,"-");

strcat(temp1,shuzu3);

strcat(temp1,".txt");

return temp1;

}

//函数名称：panduanNULL

//函数目的：判断一个句子里面有没有空的部分

int panduanNULL(ZHUWEIBIN arr)

{

//1是非空，0是空

int i;

for(i=0;i<40;i++)

{

if(arr.zhu[i]!=0)

{

return 1;

}

}

for(i=0;i<40;i++)

{

if(arr.wei[i]!=0)

{

return 1;

}

}

for(i=0;i<40;i++)

{

if(arr.bin[i]!=0)

{

return 1;

}

}

return 0;

}

//函数名称：shenmeci

//函数目的：直接查找相对应的谓语/宾语词

char\* shenmeci(ZHUWEIBIN arr)

{

int temp=0,i;

for(i=0;i<40;i++)

{

if(arr.bin[i]!=0)

{

temp=1;

}

}

if(temp==1)

{

static char tempp[50];

strcpy(tempp,weiyuci(arr));

return tempp;

//查找谓语词

}

if(temp==0)

{

static char tempp[50];

strcpy(tempp,binyuci(arr));

return tempp;

//查找宾语词

}

}

//函数名称：panduanshenmeci

//函数目的：通过一个句子里面某些部分词空不空，来进行判断

int panduanshenmeci(ZHUWEIBIN arr)

{

int temp=0,i,tteemmpp=0;

for(i=0;i<40;i++)

{

if(arr.bin[i]!=0)

{

temp=1;

}

}

for(i=0;i<40;i++)

{

if(arr.wei[i]!=0)

{

tteemmpp=1;

}

}

if(temp==0&&tteemmpp==0)

{

return 0;

//只有主语

}

if(temp==0&&tteemmpp==1)

{

return 1;

//有谓语，无宾语

}

if(temp==1&&tteemmpp==0)

{

return 2;

//有宾语，无谓语

}

if(temp==1&&tteemmpp==1)

{

return 3;

//有宾语，有谓语

}

}

void replace(char \*str,const char \*s1,const char \*s2)

{

int i;

int size1 = strlen(s1), size2 = strlen(s2);

int delta = size2 - size1;

char \*pfind = strstr(str, s1);

char \*p;

while(pfind)

{

//移动替换位置之后的字符，为替换字符串留出空间

//向前/向后移动时，p指针移动方向不同，

if(delta>0)

{

for(p=str+strlen(str);p>=pfind+size1;p--)

{

\*(p+delta)=\*p;

}

}

else if(delta<0)

{

for(p=pfind+size1;p<=str+strlen(str);p++)

{

\*(p+delta)=\*p;

}

}

for(i=0;i<size2;i++)

{

\*(pfind+i)=\*(s2+i);

}

pfind=strstr(pfind+1,s1);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

#include<graphics.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

void fish(int x,int y,float k,int color,int page)

{

int x0[103];

int y0[103];

int i;

x=x-497;

y=y-411;

x0[0]=k\*(x+497);

y0[0]=k\*(y+411);

x0[1]=k\*(x+508);

y0[1]=k\*(y+425);

x0[2]=k\*(x+511);

y0[2]=k\*(y+436);

x0[3]=k\*(x+513);

y0[3]=k\*(y+441);

x0[4]=k\*(x+515);

y0[4]=k\*(y+451);

x0[5]=k\*(x+516);

y0[5]=k\*(y+452);

x0[6]=k\*(x+515);

y0[6]=k\*(y+451);

x0[7]=k\*(x+518);

y0[7]=k\*(y+453);

x0[8]=k\*(x+515);

y0[8]=k\*(y+451);

x0[9]=k\*(x+514);

y0[9]=k\*(y+454);

x0[10]=k\*(x+513);

y0[10]=k\*(y+456);

x0[11]=k\*(x+513);

y0[11]=k\*(y+460);

x0[12]=k\*(x+508);

y0[12]=k\*(y+466);

x0[13]=k\*(x+507);

y0[13]=k\*(y+468);

x0[14]=k\*(x+503);

y0[14]=k\*(y+471);

x0[15]=k\*(x+499);

y0[15]=k\*(y+475);

x0[16]=k\*(x+494);

y0[16]=k\*(y+481);

x0[17]=k\*(x+501);

y0[17]=k\*(y+481);

x0[18]=k\*(x+506);

y0[18]=k\*(y+482);

x0[19]=k\*(x+513);

y0[19]=k\*(y+482);

x0[20]=k\*(x+517);

y0[20]=k\*(y+483);

x0[21]=k\*(x+521);

y0[21]=k\*(y+485);

x0[22]=k\*(x+528);

y0[22]=k\*(y+486);

x0[23]=k\*(x+538);

y0[23]=k\*(y+488);

x0[24]=k\*(x+546);

y0[24]=k\*(y+489);

x0[25]=k\*(x+558);

y0[25]=k\*(y+487);

x0[26]=k\*(x+574);

y0[26]=k\*(y+486);

x0[27]=k\*(x+576);

y0[27]=k\*(y+483);

x0[28]=k\*(x+581);

y0[28]=k\*(y+479);

x0[29]=k\*(x+585);

y0[29]=k\*(y+475);

x0[30]=k\*(x+589);

y0[30]=k\*(y+473);

x0[31]=k\*(x+593);

y0[31]=k\*(y+471);

x0[32]=k\*(x+595);

y0[32]=k\*(y+468);

x0[33]=k\*(x+597);

y0[33]=k\*(y+463);

x0[34]=k\*(x+599);

y0[34]=k\*(y+461);

x0[35]=k\*(x+600);

y0[35]=k\*(y+457);

x0[36]=k\*(x+601);

y0[36]=k\*(y+455);

x0[37]=k\*(x+602);

y0[37]=k\*(y+452);

x0[38]=k\*(x+604);

y0[38]=k\*(y+456);

x0[39]=k\*(x+608);

y0[39]=k\*(y+461);

x0[40]=k\*(x+610);

y0[40]=k\*(y+464);

x0[41]=k\*(x+609);

y0[41]=k\*(y+468);

x0[42]=k\*(x+611);

y0[42]=k\*(y+472);

x0[43]=k\*(x+614);

y0[43]=k\*(y+474);

x0[44]=k\*(x+617);

y0[44]=k\*(y+477);

x0[45]=k\*(x+620);

y0[45]=k\*(y+479);

x0[46]=k\*(x+623);

y0[46]=k\*(y+480);

x0[47]=k\*(x+623);

y0[47]=k\*(y+479);

x0[48]=k\*(x+623);

y0[48]=k\*(y+478);

x0[49]=k\*(x+623);

y0[49]=k\*(y+477);

x0[50]=k\*(x+623);

y0[50]=k\*(y+476);

x0[51]=k\*(x+623);

y0[51]=k\*(y+475);

x0[52]=k\*(x+623);

y0[52]=k\*(y+474);

x0[53]=k\*(x+623);

y0[53]=k\*(y+473);

x0[54]=k\*(x+623);

y0[54]=k\*(y+472);

x0[55]=k\*(x+623);

y0[55]=k\*(y+445);

x0[56]=k\*(x+623);

y0[56]=k\*(y+427);

x0[57]=k\*(x+623);

y0[57]=k\*(y+422);

x0[58]=k\*(x+623);

y0[58]=k\*(y+421);

x0[59]=k\*(x+623);

y0[59]=k\*(y+420);

x0[60]=k\*(x+623);

y0[60]=k\*(y+419);

x0[61]=k\*(x+623);

y0[61]=k\*(y+418);

x0[62]=k\*(x+623);

y0[62]=k\*(y+417);

x0[63]=k\*(x+623);

y0[63]=k\*(y+416);

x0[64]=k\*(x+623);

y0[64]=k\*(y+415);

x0[65]=k\*(x+623);

y0[65]=k\*(y+414);

x0[66]=k\*(x+623);

y0[66]=k\*(y+413);

x0[67]=k\*(x+623);

y0[67]=k\*(y+412);

x0[68]=k\*(x+623);

y0[68]=k\*(y+411);

x0[69]=k\*(x+623);

y0[69]=k\*(y+410);

x0[70]=k\*(x+621);

y0[70]=k\*(y+411);

x0[71]=k\*(x+618);

y0[71]=k\*(y+416);

x0[72]=k\*(x+616);

y0[72]=k\*(y+420);

x0[73]=k\*(x+615);

y0[73]=k\*(y+424);

x0[74]=k\*(x+612);

y0[74]=k\*(y+426);

x0[75]=k\*(x+610);

y0[75]=k\*(y+430);

x0[76]=k\*(x+608);

y0[76]=k\*(y+435);

x0[77]=k\*(x+604);

y0[77]=k\*(y+437);

x0[78]=k\*(x+603);

y0[78]=k\*(y+441);

x0[79]=k\*(x+601);

y0[79]=k\*(y+442);

x0[80]=k\*(x+599);

y0[80]=k\*(y+440);

x0[81]=k\*(x+593);

y0[81]=k\*(y+426);

x0[82]=k\*(x+592);

y0[82]=k\*(y+423);

x0[83]=k\*(x+590);

y0[83]=k\*(y+419);

x0[84]=k\*(x+588);

y0[84]=k\*(y+415);

x0[85]=k\*(x+586);

y0[85]=k\*(y+411);

x0[86]=k\*(x+583);

y0[86]=k\*(y+409);

x0[87]=k\*(x+577);

y0[87]=k\*(y+404);

x0[88]=k\*(x+571);

y0[88]=k\*(y+402);

x0[89]=k\*(x+564);

y0[89]=k\*(y+402);

x0[90]=k\*(x+554);

y0[90]=k\*(y+397);

x0[91]=k\*(x+545);

y0[91]=k\*(y+399);

x0[92]=k\*(x+528);

y0[92]=k\*(y+394);

x0[93]=k\*(x+530);

y0[93]=k\*(y+401);

x0[94]=k\*(x+515);

y0[94]=k\*(y+404);

x0[95]=k\*(x+514);

y0[95]=k\*(y+405);

x0[96]=k\*(x+511);

y0[96]=k\*(y+408);

x0[97]=k\*(x+509);

y0[97]=k\*(y+409);

x0[98]=k\*(x+506);

y0[98]=k\*(y+409);

x0[99]=k\*(x+503);

y0[99]=k\*(y+410);

x0[100]=k\*(x+501);

y0[100]=k\*(y+410);

x0[101]=k\*(x+498);

y0[101]=k\*(y+411);

x0[102]=k\*(x+497);

y0[102]=k\*(y+411);

// ѳ ʼ 㸴ԭ

for(i=0;i<103;i++)

{

x0[i]=x0[i]-(x+497)\*(k-1);

y0[i]=y0[i]-(y+411)\*(k-1);

}

Bezier(color,103,page,x0,y0,0);

fill\_1(color,page);

}

void fish\_head(int x,int y,float k,int color,int page)

{

int x0[42];

int y0[42];

int i;

x=x-497;

y=y-411;

x0[0]=k\*(x+497);

y0[0]=k\*(y+411);

x0[1]=k\*(x+493);

y0[1]=k\*(y+411);

x0[2]=k\*(x+491);

y0[2]=k\*(y+413);

x0[3]=k\*(x+489);

y0[3]=k\*(y+415);

x0[4]=k\*(x+486);

y0[4]=k\*(y+416);

x0[5]=k\*(x+483);

y0[5]=k\*(y+418);

x0[6]=k\*(x+480);

y0[6]=k\*(y+420);

x0[7]=k\*(x+479);

y0[7]=k\*(y+421);

x0[8]=k\*(x+474);

y0[8]=k\*(y+423);

x0[9]=k\*(x+471);

y0[9]=k\*(y+426);

x0[10]=k\*(x+468);

y0[10]=k\*(y+428);

x0[11]=k\*(x+466);

y0[11]=k\*(y+430);

x0[12]=k\*(x+464);

y0[12]=k\*(y+432);

x0[13]=k\*(x+462);

y0[13]=k\*(y+434);

x0[14]=k\*(x+461);

y0[14]=k\*(y+436);

x0[15]=k\*(x+459);

y0[15]=k\*(y+436);

x0[16]=k\*(x+458);

y0[16]=k\*(y+437);

x0[17]=k\*(x+454);

y0[17]=k\*(y+438);

x0[18]=k\*(x+455);

y0[18]=k\*(y+446);

x0[19]=k\*(x+456);

y0[19]=k\*(y+446);

x0[20]=k\*(x+457);

y0[20]=k\*(y+448);

x0[21]=k\*(x+460);

y0[21]=k\*(y+449);

x0[22]=k\*(x+463);

y0[22]=k\*(y+453);

x0[23]=k\*(x+465);

y0[23]=k\*(y+454);

x0[24]=k\*(x+467);

y0[24]=k\*(y+456);

x0[25]=k\*(x+468);

y0[25]=k\*(y+457);

x0[26]=k\*(x+470);

y0[26]=k\*(y+458);

x0[27]=k\*(x+473);

y0[27]=k\*(y+461);

x0[28]=k\*(x+475);

y0[28]=k\*(y+462);

x0[29]=k\*(x+477);

y0[29]=k\*(y+464);

x0[30]=k\*(x+479);

y0[30]=k\*(y+466);

x0[31]=k\*(x+482);

y0[31]=k\*(y+467);

x0[32]=k\*(x+483);

y0[32]=k\*(y+468);

x0[33]=k\*(x+484);

y0[33]=k\*(y+472);

x0[34]=k\*(x+488);

y0[34]=k\*(y+473);

x0[35]=k\*(x+490);

y0[35]=k\*(y+475);

x0[36]=k\*(x+492);

y0[36]=k\*(y+475);

x0[37]=k\*(x+493);

y0[37]=k\*(y+476);

x0[38]=k\*(x+496);

y0[38]=k\*(y+477);

x0[39]=k\*(x+498);

y0[39]=k\*(y+478);

x0[40]=k\*(x+501);

y0[40]=k\*(y+479);

x0[41]=k\*(x+507);

y0[41]=k\*(y+478);

// ѳ ʼ 㸴ԭ

for(i=0;i<42;i++)

{

x0[i]=x0[i]-(x+497)\*(k-1);

y0[i]=y0[i]-(y+411)\*(k-1);

}

Bezier(color,42,page,x0,y0,0);

//fill\_1(color,page);

}

void fish\_eye(int x,int y,float k,int color,int page)

{

int i;

x=x-495;

y=y-412;

x=k\*(x+480);

y=k\*(y+440);

x=x-(x+497)\*(k-1);

y=y-(y+411)\*(k-1);

circle\_with\_drawing(x,y,9\*k,color,page,0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：huatu1

\*\*文件目的：实现文生图的一部分绘画

\*\*时间复杂度：Bresenham复杂度

\*\*空间复杂度：Bresenham复杂度

\*/

#include<huatu1.h>

#include<Bezier.h>

#include<draw.h>

#include<stdio.h>

#include<string.h>

#include<alloc.h>

#include<stdlib.h>

#include<color.h>

/\*

#define XMAX 300

#define YMAX 300

#define XHALF 150

#define YHALF 150

\*/

void background(int x,int y,int page)

{

fill\_color(x,y,x+400,y+400,WHITE\_,page,0);

}

void yellow\_head(int x,int y,int page)

//黄种人的脸，采用橘色

{

ellipse\_(x+200,y+200,150,165,TANGERINE\_,page,0,0);

}

void brown\_head(int x,int y,int page)

//黄种人黝黑的脸，采用褐色

{

ellipse\_(x+200,y+200,150,165,BROWN\_,page,0,0);

}

void black\_head(int x,int y,int page)

//黑种人的脸，采用黑色

{

ellipse\_(x+200,y+200,150,165,BLACK\_,page,0,0);

}

void white\_head(int x,int y,int page)

//白种人的脸或黄种人白净的脸，采用粉扑桃色

{

ellipse\_(x+200,y+200,150,165,PEACH\_PUFF\_,page,0,0);

}

void black\_eye(int x,int y,int page)

{

ellipse\_(x+120,y+60,40,30,WHITE\_,page,0,0);

ellipse\_(x+280,y+60,40,30,WHITE\_,page,0,0);

circle\_with\_drawing(x+120,y+60,20,BLACK\_,page,0);

circle\_with\_drawing(x+280,y+60,20,BLACK\_,page,0);

}

void blue\_eye(int x,int y,int page)

{

ellipse\_(x+120,y+60,40,30,WHITE\_,page,0,0);

ellipse\_(x+280,y+60,40,30,WHITE\_,page,0,0);

circle\_with\_drawing(x+120,y+60,20,BLUE\_,page,0);

circle\_with\_drawing(x+280,y+60,20,BLUE\_,page,0);

}

void red\_eye(int x,int y,int page)

{

ellipse\_(x+120,y+60,40,30,WHITE\_,page,0,0);

ellipse\_(x+280,y+60,40,30,WHITE\_,page,0,0);

circle\_with\_drawing(x+120,y+60,20,RED\_,page,0);

circle\_with\_drawing(x+280,y+60,20,RED\_,page,0);

}

void nose(int x,int y,int page)

{

int x0[12];

int y0[12];

x0[0]=x+170;

y0[0]=y+140;

x0[1]=x+170;

y0[1]=y+200;

x0[2]=x+130;

y0[2]=y+210;

x0[3]=x+125;

y0[3]=y+220;

x0[4]=x+140;

y0[4]=y+230;

x0[5]=x+200;

y0[5]=y+245;

x0[6]=x+260;

y0[6]=y+230;

x0[7]=x+275;

y0[7]=y+220;

x0[8]=x+270;

y0[8]=y+210;

x0[9]=x+230;

y0[9]=y+200;

x0[10]=x+230;

y0[10]=y+140;

x0[11]=x+170;

y0[11]=y+140;

Bezier(TANGERINE\_,12,page,x0,y0,1);

fill(x+200,y+141,TANGERINE\_,page,0);

}

void smile\_mouth(int x,int y,int page)

{

int x0[5];

int y0[5];

x0[0]=x+140;

y0[0]=y+270;

x0[1]=x+180;

y0[1]=y+300;

x0[2]=x+200;

y0[2]=y+320;

x0[3]=x+220;

y0[3]=y+300;

x0[4]=x+260;

y0[4]=y+270;

Bezier(RED\_,5,page,x0,y0,1);

}

void laugh\_mouth(int x,int y,int page)

{

int x0[6];

int y0[6];

x0[0]=x+140;

y0[0]=y+270;

x0[1]=x+180;

y0[1]=y+320;

x0[2]=x+200;

y0[2]=y+340;

x0[3]=x+220;

y0[3]=y+320;

x0[4]=x+260;

y0[4]=y+270;

x0[5]=x+140;

y0[5]=y+270;

Bezier(RED\_,6,page,x0,y0,1);

fill(x+200,y+271,RED\_,page,0);

}

void sad\_mouth(int x,int y,int page)

{

int x0[5];

int y0[5];

x0[0]=x+140;

y0[0]=y+320;

x0[1]=x+180;

y0[1]=y+290;

x0[2]=x+200;

y0[2]=y+270;

x0[3]=x+220;

y0[3]=y+290;

x0[4]=x+260;

y0[4]=y+320;

Bezier(RED\_,5,page,x0,y0,1);

}

void boy\_hair(int x,int y,int page)

{

;

}

void girl\_hair(int x,int y,int page)

{

;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

/\*文件名称：hz

/\*文件目的：实现汉字的输入法

/\*时间复杂度：GB2312汉字数

/\*空间复杂度：\Theta(1)

\*/

#include<hz.h>

#include<pchat.h>

#include<page.h>

#include<AI.h>

#include<draw.h>//test

extern int MouseX,MouseY,MouseS,press;

extern void\* buffer;

extern union REGS our\_AI\_regs;

extern int MouseX\_memory,MouseY\_memory;

//函数名称：input\_method

//函数目的：汉字输入法调入

int input\_method(int x, int y, char \*str, int value, char \*py, int page, int\* choose, int \*keyboard, int\* pianyi,char\* inputinput,int counting,char input[51][421],int\* input\_counter)

{

int status = 0;

int ha,ah;

//int color[201][41];

FILE \*fp = NULL, \*oldfp = NULL;

int fJudge = FAIL;

char \*p = py;

int trigger = 1; //进入时触发输入标志

char temphz[5][3] = {{'\0', '\0', '\0'}, {'\0', '\0', '\0'}, {'\0', '\0', '\0'}, {'\0', '\0', '\0'}, {'\0', '\0', '\0'}}, temp[3];

int fposition = 0;

int hznow = 0, hznum = 0;

int asc, i;

int temptemptemp = 0;

int keykeykeyboard;

int kkk;

int jushu;

int sum\_height;

int height[100];

int PyStartx = x + 8, PyStarty = y + 4;

int HzStartx = x + 8, HzStarty = y + 22;

char \*ABpath = "C:\\CODE\\hzk\\"; //汉语拼音检索标准路径

char pypath[45]; // 汉语拼音检索相对路径

// settextjustify(LEFT\_TEXT, CENTER\_TEXT);

strcpy(pypath, "C:\\CODE\\hzk\\");

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX, MouseY, page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX, MouseY, page);

delay(20);

truly\_clean\_mouse(MouseX, MouseY, page);

//memory\_hz(x-96,y+80,color,page);

while(1)

{

//for(;;){string\_drawing("啊啊啊",0,0,24,15,9,-6,DARK\_GRAY\_,40000,120,page);}//test

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX, MouseY, page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX, MouseY, page);

delay(20);

truly\_clean\_mouse(MouseX, MouseY, page);

//箭头框

if (MouseX >= 947 && MouseX <= 1011 && MouseY >= 662 && MouseY <= 726)

{

if (mouse\_press(947, 662, 1011, 726) == 2)

{

if (status == 0)

{

MouseS = 1;

status = 5;

}

continue;

}

else if (mouse\_press(947, 662, 1011, 726) == 1)

{

fill\_color(85, 650, 935, 738, SNOW\_, page, 0);

color\_change\_chat1(5, page);

status = 5;

MouseS = 0;

temptemptemp = 0;

//执行AI功能的入口函数->后端入口

//memset(input,0,100001);

//memset(eventual,0,100001);

return 100;

}

}

//退出

if (MouseX >= 964 && MouseX <= 1024 && MouseY >= 0 && MouseY <= 60)

{

if (mouse\_press(964, 0, 1024, 60) == 2)

{

if (status == 0)

{

MouseS = 1;

status = 1;

}

continue;

}

else if (mouse\_press(964, 0, 1024, 60) == 1)

{

color\_change\_chat1(1, page);

status = 1;

MouseS = 0;

page = PAGE\_START;

change\_page(&page);

temptemptemp = 0;

}

}

//文生文按键

if (MouseX >= 85 && MouseX <= 235 && MouseY >= 585 && MouseY <= 638)

{

if (mouse\_press(85, 585, 235, 638) == 2)

{

if (status == 0)

{

MouseS = 1;

status = 3;

}

continue;

}

else if (mouse\_press(85, 585, 235, 638) == 1)

{

//color\_change\_chat1(3, page);

status = 3;

MouseS = 0;

\*choose = 0;

temptemptemp = 0;

\*input\_counter=0;

int kkk;

for(kkk=0;kkk<51;kkk++)

{

memset(input[kkk],0,421);

}

strcpy(input[0],"您好！请问有什么可以帮助您的吗？");

\*input\_counter=1;

color\_change\_chat1(3,page);

}

}

//文生图按键

if (MouseX >= 315 && MouseX <= 465 && MouseY >= 585 && MouseY <= 638)

{

if (mouse\_press(315, 585, 465, 638) == 2)

{

if (status == 0)

{

MouseS = 1;

status = 4;

}

continue;

}

else if (mouse\_press(315, 585, 465, 638) == 1)

{

//color\_change\_chat1(4, page);

status = 4;

MouseS = 0;

\*choose = 1;

temptemptemp = 0;

\*input\_counter=0;

for(kkk=0;kkk<51;kkk++)

{

memset(input[kkk],0,421);

}

color\_change\_chat1(4,page);

}

}

if (trigger || kbhit())

{ // 第一次进入自动触发 以后均需键盘

//clrmous(MouseX,MouseY);//test

//mouseinit();//test

//UpdateMouse();

trigger = 0;

if (kbhit())

{

value = bioskey(0);

}

asc = value & 0xff;

/\*特殊按键处理\*/

switch (value)

{

case UP:

//对话框

//if((MouseX>=0&&MouseX<=963&&MouseY>=0&&MouseY<=575)||

//(MouseX>=964&&MouseX<=1024&&MouseY>=61&&MouseY<=575))

//{

//if(mouse\_press(0,1024,0,575)==1)

//{

jushu = text\_jushu(input);

sum\_height = 0;

for (i = 0; input[i][0] != '\0'; i++)

{

sum\_height += strlen(input[i]) / 46;

sum\_height += 2;

}

\*keyboard -= 2;

if (sum\_height >= 15)

{

display\_text(input, \*pianyi, page, \*keyboard);

}

// }

//}

break;

case DOWN:

//if((MouseX>=0&&MouseX<=963&&MouseY>=0&&MouseY<=575)||

//(MouseX>=964&&MouseX<=1024&&MouseY>=61&&MouseY<=575))

//{

//if(mouse\_press(0,1024,0,575)==1)

//{

jushu=text\_jushu(input);

sum\_height = 0;

for (i = 0; input[i][0] != '\0'; i++)

{

sum\_height+=strlen(input[i])/46;

sum\_height+=2;

}

\*keyboard+=2;

if (sum\_height >= 15)

{

display\_text(input,\*pianyi,page,\*keyboard);

}

//}

//}

break;

case BACK:

p--;

\*p='\0';

if(inputinput[0]!=0)

{

if(inputinput[strlen(inputinput)-1]<0)

{

fill\_color(x-94,y+42,x-62,y+74,SNOW\_,page,0);

fill\_color(x-64,y+72,x+172,y+126,SNOW\_,page,0);

}

else if(inputinput[strlen(inputinput)-1]>0)

{

fill\_color(x-90,y+42,x-62,y+74,SNOW\_,page,0);

fill\_color(x-64,y+72,x+172,y+126,SNOW\_,page,0);

}

}

else

{

;

}

if(py[0]=='\0')

{

str[0]='\0';

if(oldfp)

{

fclose(oldfp);

}

if(fp)

{

fclose(fp);

}

return 3;

}

return 4;

case SPACE:

strcpy(str,temphz[hznow]);

if(oldfp)

fclose(oldfp);

if(fp)

fclose(fp);

//clean\_hz(x-96,y+80,color,page);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

return 1;

case ENTER:

strcpy(str, py);

if (oldfp)

fclose(oldfp);

if (fp)

fclose(fp);

//clean\_hz(x-96,y+80,color,page);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

return 666;

case LASTLINE:

if (fposition >= 8)

{ //接下来重定位文件指针前八个字节（四个汉字）

fposition -= 8;

}

break;

case NEXTLINE:

if (!feof(fp))

{ //接下来重定位文件指针后八个字节（四个汉字）

fposition += 8;

}

break;

case LEFT://左移动一个

//if(hznow)

//{

//hznow--;

if (fposition >= 8)

{

fposition -= 8;

//hznow=3;

}

//}

//else if(fposition>=8)//需要左换页

//{

// fposition-=8;

// hznow=3;

//}

break;

case RIGHT:

if (hznow < hznum - 1)

{ //同左

//hznow++;

if (!feof(fp))

{

fposition += 8;

//hznow=0;

}

}

//else if(!feof(fp))

//{

// fposition+=8;

// hznow=0;

//}

break;

/\*按数字键选中输入汉字\*/

case FIRST:

if (py[0] == '\0')

{

;

}

else

{

strcpy(str, temphz[0]);

if (oldfp)

fclose(oldfp);

if (fp)

fclose(fp);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

//string\_drawing("Hello world!",0,0,24,15,9,-6,BLACK\_,40000,120,page);

return 1;

}

case SECOND:

if (py[0] == '\0')

{

;

}

else

{

strcpy(str,temphz[1]);

if(oldfp)

fclose(oldfp);

if(fp)

fclose(fp);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

//string\_drawing("Hello world!",0,0,24,15,9,-6,BLACK\_,40000,120,page);

return 1;

}

case THIRD:

if (py[0] == '\0')

{

;

}

else

{

strcpy(str,temphz[2]);

if(oldfp)

fclose(oldfp);

if(fp)

fclose(fp);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

//string\_drawing("Hello world!",0,0,24,15,9,-6,BLACK\_,40000,120,page);

return 1;

}

case FOURTH:

if (py[0] == '\0')

{

;

}

else

{

strcpy(str, temphz[3]);

if (oldfp)

fclose(oldfp);

if (fp)

fclose(fp);

fill\_color(x - 64, y + 70, x + 138, y + 124, SNOW\_, page, 0);

//string\_drawing("Hello world!",0,0,24,15,9,-6,BLACK\_,40000,120,page);

return 1;

}

}

/\*输入字符处理\*/

if (asc > 31 && asc < 127 && strlen(py) < MAXPY && asc != '[' && asc != ']')

{

//有效输入时则复位

\*p = asc;

p++;

fposition = 0;

hznow = 0;

}

pyFrm(x, y, x + 200, y + 40, page);

string\_drawing(py, PyStartx - 64, PyStarty + 72, 24, 15, 9, -6, DARK\_GRAY\_, 40000, 120, page, 0);

//setfillstyle(1,WHITE);

//settextstyle(1,0,2);

//settextjustify(LEFT\_TEXT,CENTER\_TEXT);

//outtextxy(PyStartx,PyStarty,py);//拼音字体

//string\_drawing(PyStartx,PyStarty,py,16,BLUE\_,page);

strcat(pypath, py);

strcat(pypath, ".txt");

if (fJudge)

{ //将当前文件指针保存 同时关闭上一个文件 为输入特殊字符准备

if (oldfp)

{

fclose(oldfp);

}

oldfp = fp;

}

//string\_drawing(pypath,0,0,48,26,15,0,RED\_,10000,10000,page);//test

if ((fp = fopen(pypath, "r")) == NULL)

{ //特殊字符存在 保留上一个文件检索结果

fJudge = FAIL;

fp = oldfp;

}

else

{

fJudge = SUCCESS;

}

if (fp)

{

fseek(fp, fposition, SEEK\_SET);

for (i = 0; i < 5; i++)

{

fread(temphz[i], 2, 1, fp); //读入一个汉字

if (feof(fp))

{ //读到文件尾

hznum = i; //按道理此处文件尾多读一次 需要减一 然而此处不减一的效果更好

break;

}

}

if (!feof(fp))

{ //未读到文件尾 全显汉字

hznum = 4;

}

//string\_drawing(temphz[hznow],HzStartx+hznow\*50-68,HzStarty+20,24,15,9,-6,DARK\_GRAY\_,400,12,page);

for (i = 0; i < hznum; i++)

{

//setcolor(BLUE);

//settextstyle(1,0,2);

//xouttextxy(HzStartx + i \* 50, HzStarty + 5, itostr(i + 1, temp), DARK\_GRAY\_);

//string\_drawing(HzStartx+i\*50,HzStarty+2,itostr(i+1,temp),16,DARK\_GRAY\_,page);

//string\_drawing(itostr(i+1,temp),0,0,16,8,6,-3,DARK\_GRAY\_,400,12,page);//test

string\_drawing(itostr(i + 1, temp), HzStartx + i \* 50 - 64, HzStarty + 76, 24, 15, 9, -6, DARK\_GRAY\_, 40000, 120, page, 0);

// puthz(HzStartx+i\*50+16,HzStarty,temphz[i],16,16,DARK\_GRAY\_);

//string\_drawing(HzStartx+i\*50+16,HzStarty,temphz[i],16,DARK\_GRAY\_,page);

string\_drawing(temphz[i], HzStartx + i \* 50 - 48, HzStarty + 76, 24, 15, 9, -6, DARK\_GRAY\_, 40000, 120, page, 0);

}

//puthz(HzStartx+hznow\*50+16,HzStarty,temphz[hznow],16,16,CYAN);//显示选中汉字

//string\_drawing(HzStartx+hznow\*50+16,HzStarty,temphz[hznow],16,DARK\_GRAY\_,page);

//string\_drawing(temphz[hznow],HzStartx+hznow\*50-64,HzStarty+16,32,20,12,-8,BLACK\_,40000,120,page);

}

}

//string\_drawing(pypath,0,0,24,15,9,-6,DARK\_GRAY\_,400,12,page);//test

strcpy(pypath, ABpath); //绝对路径复原（不可少）

value = 0;

//hznow=0;

//mouse\_drawing(MouseX,MouseY,page);

//drawtext(HzStartx + hznow \* 50 + 16, HzStarty, temphz[hznow], 16, CYAN);

delay(20);

}

}

//函数名称：itostr

//函数目的：数字标号

char \*itostr(int a, char \*s)

{

switch (a)

{

case 1:

strcpy(s, "1");

return s;

case 2:

strcpy(s, "2");

return s;

case 3:

strcpy(s, "3");

return s;

case 4:

strcpy(s, "4");

return s;

}

return s;

}

//函数名称：pyFrm

//函数目的：输入法小框

void pyFrm(int x1, int y1, int x2, int y2, int page)

{

//fill\_color(x1,y1,x2,y2,WHITE\_,page);

//fill\_color(x1,y1-32,x2,y2-32,DARK\_GRAY\_,page);

fill\_color(x1 - 64, y1 + 72, x2 - 64, y2 + 80, WHITE\_, page, 0);

line\_drawing(x1 - 59, y1 + 92, x2 - 69, y1 + 92, BLUE\_, page, 0);

}

//函数名称：memory\_hz

//函数目的：记忆输入法之前的画面

void memory\_hz(int x, int y, int color[201][41], int page)

{

int i, j;

for (i = 0; i < 201; i++)

{

for (j = 0; j < 41; j++)

{

color[i][j] = get\_pixel\_color(x + i, y + j, page);

}

}

}

//函数名称：clean\_hz

//函数目的：实现输入法画面的清除

void clean\_hz(int x, int y, int color[201][41], int page)

{

int i, j;

for (i = 0; i < 201; i++)

{

for (j = 0; j < 41; j++)

{

point\_drawing(x + i, y + j, color[i][j], page);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

/\*文件名称：input

/\*文件目的：实现字符在界面上的输入

/\*时间复杂度：正比于牵扯到的像素个数

/\*空间复杂度：OH(1)

\*/

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<draw.h>

#include<color.h>

#include<page.h>

#include<mouse.h>

#include<dos.h>

#include<time.h>

#include<bios.h>

#include<xiezi.h>

#include<pstat.h>

#include<English.h>

#include<pregi.h>

#include<input.h>

#include<conio.h>

//函数名称：visible\_input

//函数目的：实现可视输入

int visible\_input(char \*str,int x,int y,int limit,int color,int color2,int page)

{

//clrmous(MouseX,MouseY,page);

int sum=0;

char keyboard;

//line\_drawing(x+10,y+6,x+10,y+20,color,page);//光标

//不画第一个

while(1)

{

keyboard=bioskey(0);

if(sum<limit)

{

fill\_color(550,175,550+48\*6,175+48,SPRING\_GREEN\_,page,0);

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)//换行、空格、退出

{

if(keyboard!='\b')

{

if((keyboard>='0'&&keyboard<='9')||(keyboard>='A'&&keyboard<='Z')||(keyboard>='a'&&keyboard<='z'))

{

//改变字符串的值

\*(str+sum)=keyboard;

\*(str+sum+1)='\0';

//改变光标位置

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,SPRING\_GREEN\_,page);

English\_text\_drawing(x+8+sum\*18,y-1,str+sum,color,page,0);

//outtextxy(x+8+sum\*18,y-1,str+sum);

sum++;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

}

else

{

return ILLEGAL\_INPUT;

//啥也不干

}

}

else if(keyboard=='\b'&&sum>0)

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+32,DARK\_GRAY\_,page,0);

//清除上一个输入的字符

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+36,color2,page,0);

sum--; //改变sum值

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

//重置光标

\*(str+sum)='\0';

//改变字符串

}

}

else

{

fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color2,page,0);

break;

}

}

else if(sum>=limit)

{

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)

{

string\_drawing("最多十五位！",550,175,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

while(1)

{

if(kbhit())

{

char value=bioskey(0);

if(value==13)

{

return LEGAL\_INPUT;

}

else if(value=='\b')

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+54,DARK\_GRAY\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

break;

}

}

}

}

else

{

if(keyboard=='\b'&&sum>0)

{

//跟前面的退格一模一样

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+32,DARK\_GRAY\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

}

else

{

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color,page,0);

break;

}

}

}

}

return LEGAL\_INPUT;

}

//函数名称：invisible\_input

//函数目的：实现函数的不可视输入

int invisible\_input(char \*str,int x,int y,int limit,int color,int color2,int page)

{

//clrmous(MouseX,MouseY,page);

int sum=0;

char keyboard;

//line\_drawing(x+10,y+6,x+10,y+20,color,page);//光标

//不画第一个

while(1)

{

keyboard=bioskey(0);

if(sum<limit)

{

fill\_color(550,175,550+48\*6,175+48,SPRING\_GREEN\_,page,0);

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)//换行、空格、退出

{

if(keyboard!='\b')

{

if((keyboard>='0'&&keyboard<='9')||(keyboard>='A'&&keyboard<='Z')||(keyboard>='a'&&keyboard<='z'))

{

//改变字符串的值

\*(str+sum)=keyboard;

\*(str+sum+1)='\0';

//改变光标位置

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,SPRING\_GREEN\_,page);

English\_text\_drawing(x+8+sum\*18,y+4,"-",color,page,0);

//outtextxy(x+8+sum\*18,y-1,str+sum);

sum++;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

}

else

{

return ILLEGAL\_INPUT;

//啥也不干

}

}

else if(keyboard=='\b'&&sum>0)

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+40,DARK\_GRAY\_,page,0);

//清除上一个输入的字符

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+36,color2,page,0);

sum--; //改变sum值

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

//重置光标

\*(str+sum)='\0';

//改变字符串

}

}

else

{

fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+40,color2,page,0);

break;

}

}

else if(sum>=limit)

{

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)

{

string\_drawing("最多十五位！",550,175,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

while(1)

{

if(kbhit())

{

char value=bioskey(0);

if(value==13)

{

return LEGAL\_INPUT;

}

else if(value=='\b')

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+64,DARK\_GRAY\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

break;

}

}

}

}

else

{

if(keyboard=='\b'&&sum>0)

{

//跟前面的退格一模一样

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+32,DARK\_GRAY\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

}

else

{

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color,page,0);

break;

}

}

}

}

return LEGAL\_INPUT;

}

//函数名称：visible\_input1

//函数目的：实现可视输入

int visible\_input1(char \*str,int x,int y,int limit,int color,int color2,int page)

{

//clrmous(MouseX,MouseY,page);

int sum=0;

char keyboard;

//line\_drawing(x+10,y+6,x+10,y+20,color,page);//光标

//不画第一个

while(1)

{

keyboard=bioskey(0);

if(sum<limit)

{

fill\_color(550,20,550+48\*6,68,SPRING\_GREEN\_,page,0);

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)//换行、空格、退出

{

if(keyboard!='\b')

{

if((keyboard>='0'&&keyboard<='9')||(keyboard>='A'&&keyboard<='Z')||(keyboard>='a'&&keyboard<='z'))

{

//改变字符串的值

\*(str+sum)=keyboard;

\*(str+sum+1)='\0';

//改变光标位置

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,SPRING\_GREEN\_,page);

English\_text\_drawing(x+8+sum\*18,y-1,str+sum,color,page,0);

//outtextxy(x+8+sum\*18,y-1,str+sum);

sum++;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

}

else

{

return ILLEGAL\_INPUT;

//啥也不干

}

}

else if(keyboard=='\b'&&sum>0)

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+40,GAINSBORO\_,page,0);

//清除上一个输入的字符

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+36,color2,page,0);

sum--; //改变sum值

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

//重置光标

\*(str+sum)='\0';

//改变字符串

}

}

else

{

fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color2,page,0);

break;

}

}

else if(sum>=limit)

{

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)

{

string\_drawing("最多十五位！",550,20,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

while(1)

{

if(kbhit())

{

char value=bioskey(0);

if(value==13)

{

return LEGAL\_INPUT;

}

else if(value=='\b')

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+44,GAINSBORO\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

break;

}

}

}

}

else

{

if(keyboard=='\b'&&sum>0)

{

//跟前面的退格一模一样

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+40,GAINSBORO\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

}

else

{

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color,page,0);

break;

}

}

}

}

return LEGAL\_INPUT;

}

//函数名称：invisible\_input

//函数目的：实现函数的不可视输入

int invisible\_input1(char \*str,int x,int y,int limit,int color,int color2,int page)

{

//clrmous(MouseX,MouseY,page);

int sum=0;

char keyboard;

//line\_drawing(x+10,y+6,x+10,y+20,color,page);//光标

//不画第一个

while(1)

{

keyboard=bioskey(0);

if(sum<limit)

{

fill\_color(550,20,550+48\*6,68,SPRING\_GREEN\_,page,0);

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)//换行、空格、退出

{

if(keyboard!='\b')

{

if((keyboard>='0'&&keyboard<='9')||(keyboard>='A'&&keyboard<='Z')||(keyboard>='a'&&keyboard<='z'))

{

//改变字符串的值

\*(str+sum)=keyboard;

\*(str+sum+1)='\0';

//改变光标位置

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,SPRING\_GREEN\_,page);

English\_text\_drawing(x+8+sum\*18,y+4,"-",color,page,0);

//outtextxy(x+8+sum\*18,y-1,str+sum);

sum++;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

}

else

{

return ILLEGAL\_INPUT;

//啥也不干

}

}

else if(keyboard=='\b'&&sum>0)

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+44,GAINSBORO\_,page,0);

//清除上一个输入的字符

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+36,color2,page,0);

sum--; //改变sum值

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

//重置光标

\*(str+sum)='\0';

//改变字符串

}

}

else

{

fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+44,color2,page,0);

break;

}

}

else if(sum>=limit)

{

if(keyboard!='\n'&&keyboard!='\r'&&keyboard!=' '&&keyboard!=0x1b)

{

string\_drawing("最多十五位！",550,20,48,26,15,0,RED\_,BIG\_NUMBER,BIG\_NUMBER,page,0);

while(1)

{

if(kbhit())

{

char value=bioskey(0);

if(value==13)

{

return LEGAL\_INPUT;

}

else if(value=='\b')

{

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+44,GAINSBORO\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

break;

}

}

}

}

else

{

if(keyboard=='\b'&&sum>0)

{

//跟前面的退格一模一样

fill\_color(x-6+sum\*18,y+3,x+12+sum\*18,y+44,GAINSBORO\_,page,0);

//fill\_color(x-10+sum\*18,y+3,x+7+sum\*18,y+24,color,page,0);

sum--;

//line\_drawing(x+10+sum\*18,y+6,x+10+sum\*18,y+20,color,page);

\*(str+sum)='\0';

}

else

{

//fill\_color(x+8+sum\*18,y+3,x+12+sum\*18,y+24,color,page,0);

break;

}

}

}

}

return LEGAL\_INPUT;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 函数名称：calculate

\*\* 函数目的：实现各种复杂的数学计算

\*\* 时间复杂度：最优复杂度

\*\* 空间复杂度：最优复杂度（高精度下）

\*/

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<ctype.h>

#include<jisuan.h>

// 反转字符串

void reverse(char \*str)

{

int len=strlen(str);

for (int i=0;i<len/2;i++)

{

char temp=str[i];

str[i]=str[len-i-1];

str[len-i-1]=temp;

}

}

// 比较两个无符号数的大小

int compare(char \*num1,char \*num2)

{

int len1=strlen(num1);

int len2=strlen(num2);

if (len1!=len2)

{

return len1-len2;

}

return strcmp(num1,num2);

}

// 高精度加法（无符号数）

void add(char \*num1,char \*num2,char \*result)

{

int len1=strlen(num1);

int len2=strlen(num2);

int carry=0;

int i=0;

while(i<len1||i<len2||carry)

{

int digit1=(i<len1)?(num1[i]-'0'):0;

int digit2=(i<len2)?(num2[i]-'0'):0;

int sum=digit1+digit2+carry;

result[i]=(sum%10)+'0';

carry=sum/10;

i++;

}

result[i]='\0';

reverse(result);

}

// 高精度减法（无符号数，假设 num1 >= num2）

void subtract(char \*num1,char \*num2,char \*result)

{

int len1=strlen(num1);

int len2=strlen(num2);

int borrow=0;

int i=0;

while(i<len1)

{

int digit1=num1[i]-'0';

int digit2=(i<len2)?(num2[i]-'0'):0;

int diff=digit1-digit2-borrow;

if (diff<0)

{

diff+=10;

borrow=1;

}

else

{

borrow=0;

}

result[i]=diff+'0';

i++;

}

// 去除前导零

while(i>1&&result[i-1]=='0')

{

result[i-1]='\0';

i--;

}

result[i]='\0';

reverse(result);

}

// 处理实数高精度加法，考虑正负号

void real\_add(char \*num1,char \*num2,char \*result)

{

int sign1=(num1[0]=='-')?-1:1;

int sign2=(num2[0]=='-')?-1:1;

if(num1[0]=='-')

num1++;

if(num2[0]=='-')

num2++;

// 分离整数部分和小数部分

char \*dot1=strchr(num1,'.');

char \*dot2=strchr(num2,'.');

char int\_part1[1000]={0},frac\_part1[1000]={0};

char int\_part2[1000]={0},frac\_part2[1000]={0};

char frac\_result[1000]={0},int\_result[1000]={0};

if(dot1)

{

strncpy(int\_part1,num1,dot1-num1);

strcpy(frac\_part1,dot1+1);

}

else

{

strcpy(int\_part1,num1);

}

if(dot2)

{

strncpy(int\_part2,num2,dot2-num2);

strcpy(frac\_part2,dot2+1);

}

else

{

strcpy(int\_part2,num2);

}

// 补齐小数部分长度

int max\_frac\_len=strlen(frac\_part1)>strlen(frac\_part2)?strlen(frac\_part1):strlen(frac\_part2);

while(strlen(frac\_part1)<max\_frac\_len)

{

strcat(frac\_part1,"0");

}

while(strlen(frac\_part2)<max\_frac\_len)

{

strcat(frac\_part2,"0");

}

// 反转小数部分以便从低位开始相加

reverse(frac\_part1);

reverse(frac\_part2);

// 处理同号相加

if(sign1==sign2)

{

add(frac\_part1,frac\_part2,frac\_result);

reverse(frac\_result);

// 处理小数部分进位

int carry=0;

if (strlen(frac\_result)>max\_frac\_len)

{

carry= 1;

frac\_result[strlen(frac\_result)-1]='\0';

}

// 反转整数部分以便从低位开始相加

reverse(int\_part1);

reverse(int\_part2);

if(carry)

{

char carry\_str[2]="1";

add(int\_part1,carry\_str,int\_part1);

}

add(int\_part1,int\_part2,int\_result);

reverse(int\_result);

// 合并结果

if (sign1==-1)

{

strcpy(result, "-");

}

else

{

result[0]='\0';

}

strcat(result, int\_result);

if(strlen(frac\_result) > 0)

{

strcat(result,".");

strcat(result, frac\_result);

}

}

else

{ // 异号相减

int cmp=compare(int\_part1,int\_part2);

if (cmp==0)

{

cmp=compare(frac\_part1,frac\_part2);

}

if (cmp>=0)

{

subtract(frac\_part1,frac\_part2,frac\_result);

reverse(frac\_result);

reverse(int\_part1);

reverse(int\_part2);

subtract(int\_part1,int\_part2,int\_result);

reverse(int\_result);

if (sign1==-1)

{

strcpy(result,"-");

}

else

{

result[0]='\0';

}

strcat(result, int\_result);

if (strlen(frac\_result) > 0)

{

strcat(result,".");

strcat(result,frac\_result);

}

}

else

{

subtract(frac\_part2,frac\_part1,frac\_result);

reverse(frac\_result);

reverse(int\_part2);

reverse(int\_part1);

subtract(int\_part2,int\_part1,int\_result);

reverse(int\_result);

if (sign2==-1)

{

strcpy(result,"-");

}

else

{

result[0]='\0';

}

strcat(result,int\_result);

if (strlen(frac\_result) > 0)

{

strcat(result,".");

strcat(result,frac\_result);

}

}

}

}

// 高精度乘法（无符号整数）

void multiply(char \*num1,char\*num2,char \*result)

{

int len1=strlen(num1);

int len2=strlen(num2);

int \*prod=(int \*)calloc(len1+len2,sizeof(int));

for (int i=len1-1;i>=0;i--)

{

for (int j=len2-1;j>=0;j--)

{

int mul=(num1[i]-'0')\*(num2[j]-'0');

int sum=mul+prod[i+j+1];

prod[i+j+1]=sum%10;

prod[i+j]+=sum/10;

}

}

i=0;

while(i<len1+len2&&prod[i]==0)

i++;

if(i==len1+len2)

{

result[0]='0';

result[1]='\0';

}

else

{

int j=0;

for(;i<len1+len2;i++)

{

result[j++]=prod[i]+'0';

}

result[j]='\0';

}

free(prod);

}

// 处理实数高精度乘法，考虑正负号

void real\_multiply(char \*num1, char \*num2, char \*result)

{

int sign1=(num1[0]=='-')?-1:1;

int sign2=(num2[0]=='-')?-1:1;

if (num1[0]=='-')

num1++;

if (num2[0]=='-')

num2++;

// 分离整数部分和小数部分

char \*dot1=strchr(num1,'.');

char \*dot2=strchr(num2,'.');

char int\_part1[1000]={0},frac\_part1[1000]={0};

char int\_part2[1000]={0},frac\_part2[1000]={0};

char temp\_result[2000]={0};

if(dot1)

{

strncpy(int\_part1,num1,dot1-num1);

strcpy(frac\_part1,dot1+1);

}

else

{

strcpy(int\_part1,num1);

}

if(dot2)

{

strncpy(int\_part2,num2,dot2-num2);

strcpy(frac\_part2,dot2+1);

}

else

{

strcpy(int\_part2,num2);

}

// 去掉小数点，当成整数相乘

char combined\_num1[2000],combined\_num2[2000];

strcpy(combined\_num1,int\_part1);

strcat(combined\_num1,frac\_part1);

strcpy(combined\_num2,int\_part2);

strcat(combined\_num2,frac\_part2);

multiply(combined\_num1,combined\_num2,temp\_result);

// 确定小数点位置

int frac\_digits=strlen(frac\_part1)+strlen(frac\_part2);

int result\_len=strlen(temp\_result);

if (frac\_digits>0)

{

if (frac\_digits>=result\_len)

{

char temp[2000];

strcpy(temp,"0.");

for (int i = 0;i<frac\_digits-result\_len;i++)

{

strcat(temp,"0");

}

strcat(temp,temp\_result);

strcpy(temp\_result, temp);

}

else

{

char temp[2000];

strncpy(temp, temp\_result,result\_len-frac\_digits);

temp[result\_len - frac\_digits]='\0';

strcat(temp,".");

strcat(temp,temp\_result+result\_len-frac\_digits);

strcpy(temp\_result,temp);

}

}

// 处理结果符号

if (sign1\*sign2==-1)

{

strcpy(result,"-");

strcat(result,temp\_result);

}

else

{

strcpy(result,temp\_result);

}

}

// 计算字符串表示的实数的倒数

void reciprocal(char \*input,char \*output)

{

// 将输入字符串转换为双精度浮点数

double num=strtod(input,NULL);

// 检查输入是否为 0

if (num==0)

{

strcpy(output,"倒数无意义，输入不能为 0");

}

else

{

// 计算倒数

double result=1.0/num;

// 将结果转换为字符串

sprintf(output,"%lf",result);

}

}

//判断一个数是否是素数

int isPrime(long long int num)

{

if(num<=1)

{

return 0;

}

for (int i=2;i<num;i++)

{

if (num%i==0)

{

return 0;

}

}

return 1;

}

// 计算最大公因数

long long int gcd(long long int a,long long int b)

{

while(b!=0)

{

int temp=b;

b=a%b;

a=temp;

}

return a;

}

//最小公倍数就直接用这个算就行了

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：lasthao

\*\*文件目的：实现最终的标号完成

\*\*时间复杂度：k1\*词数+k2\*字数

\*\*空间复杂度：相对较为复杂，但为最低

\*/

#include<stdio.h>

#include<lasthao.h>

#include<string.h>

#include<stdlib.h>

#include<alloc.h>

#include<stce.h>

#include<dos.h>

#include<num.h>

#include<trans12.h>

/\*AC自动机

ZHUWEIBIN\* lasthao(char\* biaohao,char\* input)

{

//基本思路：生成抽象的主谓宾的格式，进行算法封装

ZHUWEIBIN\* record=(ZHUWEIBIN\*)malloc(5\*sizeof(struct ZHUWEIBIN\*));//留够5个逻辑关系

//开堆区空间的声明最后的存储变量

char\* temp=(char\*)malloc(500);

//建立AC自动机的字典树，到后面执行效率会高些

TrieNode\* tree1=(TrieNode\*)malloc(600);

tree1=init\_automation();

TrieNode\* tree2=(TrieNode\*)malloc(600);

tree2=init\_automation\_2();

int counting=0;

int i;

Tense tense\_,already\_;

int beibei;

int dijiju=0;

for(i=0;input[i]!='\0';i++)

//直到到input的\0为止

//冷知识：空格的ASC2码是32，回车是13，EOF是26，都不是0

{

if(input[i]==','||input[i]=='?')

{

counting=0;

if(temp[0]!='\0')

{

//执行记录！往那个结构体里面储存

//先判断相关的情况

tense\_=tense(temp,tree1);

already\_=already(temp,tree2);

beibei=beidong(temp);

yijuhua(temp,tense\_,already\_,beibei,record+(dijiju++));//小心bug

}

memset(temp,0,sizeof(temp));

}

else

{

temp[counting++]=input[i];

}

}

if(input[strlen(input)-1]!=','&&input[strlen(input)-1]!='?')

{

tense\_=tense(temp,tree1);

already\_=already(temp,tree2);

beibei=beidong(temp);

//yijuhua(temp,tense\_,already\_,beibei,record+(dijiju++));//小心bug

}

free(tree1);

free(tree2);

return record;

}\*/

void lasthao(char far\* input,char far\* fencihou,ZHUWEIBIN\* record)

{

//基本思路：生成抽象的主谓宾的格式，进行算法封装

//ZHUWEIBIN\* record=(ZHUWEIBIN\*)malloc(10\*sizeof(struct ZHUWEIBIN\*));//留够5个逻辑关系

//开堆区空间的声明最后的存储变量

char\* temp=(char\*)malloc(300);

int i=0;

memset(temp,0,100);

int j;

for(i=0;i<10;i++)

{

for(j=0;j<30;j++)

{

record[i].zhu[j]=0;

record[i].wei[j]=0;

record[i].bin[j]=0;

}

}

//建立AC自动机的字典树，到后面执行效率会高些

//TrieNode\* tree1=(TrieNode\*)malloc(600);

//tree1=init\_automation();

//TrieNode\* tree2=(TrieNode\*)malloc(600);

//tree2=init\_automation\_2();

int counting=0;

Tense tense\_,already\_;

int beibei;

int dijiju=0;

for(i=0;input[i]!='\0';i++)

//直到到input的\0为止

//冷知识：空格的ASC2码是32，回车是13，EOF是26，都不是0

{

if(input[i]==','||input[i]=='?')

{

counting=0;

if(temp[0]!='\0')

{

//执行记录！往那个结构体里面储存

//先判断相关的情况

tense\_=tense(temp);

already\_=already(temp);

beibei=beidong(temp);

yijuhua(temp,tense\_,already\_,beibei,record+(dijiju++));//小心bug

}

memset(temp,0,300);

}

else

{

temp[counting++]=input[i];

}

}

if(input[strlen(input)-1]!=','&&input[strlen(input)-1]!='?')

{

tense\_=tense(temp);

already\_=already(temp);

beibei=beidong(temp);

yijuhua(fencihou,tense\_,already\_,beibei,record+(dijiju++));//小心bug

}

//strcpy(biaohao,fencihou);

//strcpy(biaohao,"牛啊");//test

free(temp);

//free(tree1);

//free(tree2);

return;

}

void yijuhua(char\* temp,Tense tense\_,Tense already\_,int beibei,ZHUWEIBIN\* zhuweibin)

{

//先给二维数组动态分配内存吧！！！

//char\*\* arr=(char\*\*)malloc(10\*sizeof(char\*));

//提示一下，第一维度是第几个词，第二维度是这个词的第几个字符

int i,j;

char chayuan[7][50]={{0}};

/\*if(arr==NULL)

{

printf("内存分配失败了！");

delay(2000);

return;

}

for(i=0;i<10;i++)

{

arr[i]=(char\*)malloc(50\*sizeof(char));

if(arr[i]==NULL)

{

printf("内存分配失败了！");

return;

}

}\*/

char arr[7][50]={{0}};

/\*for(i=0;i<10;i++)

{

memset(arr[i],0,50);

}\*/

//然后干正事

int counting\_2=0;

int count\_1=0;

//for(i=0;temp[i]!='\0';i++)

//就是说扫描的temp必须是有东西的

{

for(j=0;temp[j]!='\0';j++)

//就是说扫描的过程中，那个字符不是终止字符

{

if(temp[j]=='|')

{

counting\_2=0;

if(arr[count\_1][0]!='\0')

{

count\_1++;

}

}

else

{

arr[count\_1][counting\_2]=temp[j];

counting\_2++;

}

}

}

//zhuweibin有且仅有一个，phr可是可以有很多的

//根据分出来的词，进行这句话的分析~

//先在PHR里面进行

//PHR\* phr=(PHR\*)malloc(10\*sizeof(PHR));

//for(i=0;i<10;i++)

//{

// phr->str=(char\*)malloc(50);

//}

//phr=remark(arr);

//for(i=0;i<7;i++)

for(i=0;i<7;i++)

{

trans12(arr[i],chayuan[i]);

}

int qinghua[50]={0};

int huake[50]={0};

int wuda[50]={0};

char temppp[50]={0};

int counttt=0;

int mxz=0;

for(i=0;i<7;i++)

{

for(j=0;j<50;j++)

{

if(chayuan[i][j]=='.'||chayuan[i][j]=='\0')

{

qinghua[i]=atoi(temppp);

memset(temppp,0,50);

break;

}

else

{

temppp[j]=chayuan[i][j];

}

}

}

for(i=0;i<7;i++)

{

for(j=0;j<50;j++)

{

if(chayuan[i][j]=='\0')

{

huake[i]=atoi(temppp);

memset(temppp,0,50);

break;

}

else if(chayuan[i][j]=='.'&&mxz==0)

{

mxz=1;

counttt=j;

}

else if(chayuan[i][j]=='.'&&mxz==1)

{

huake[i]=atoi(temppp);

memset(temppp,0,50);

break;

}

else if(mxz==0)

{

;

}

else

{

temppp[j-counttt-1]=chayuan[i][j];

}

}

mxz=0;

}

for(i=0;i<7;i++)

{

for(j=0;j<50;j++)

{

if(chayuan[i][j]=='\0')

{

wuda[i]=atoi(temppp);

memset(temppp,0,50);

break;

}

else if(chayuan[i][j]=='.'&&mxz==0)

{

mxz=1;

}

else if(chayuan[i][j]=='.'&&mxz==1)

{

mxz=2;

counttt=j;

}

else if(chayuan[i][j]=='.'&&mxz==2)

{

wuda[i]=atoi(temppp);

memset(temppp,0,50);

break;

}

else if(mxz==0||mxz==1)

{

;

}

else

{

temppp[j-counttt-1]=chayuan[i][j];

}

}

mxz=0;

}

//没错，phr是数组！是数组！是数组！

//再往真实的结构里面腾东西

switch(tense\_)

{

case TENSE\_PAST:

zhuweibin->wei[0]=TENSE\_PAST;

break;

case TENSE\_PRESENT:

zhuweibin->wei[0]=TENSE\_PRESENT;

break;

case TENSE\_FUTURE:

zhuweibin->wei[0]=TENSE\_FUTURE;

break;

}

switch(already\_)

{

case TENSE\_PAST:

zhuweibin->wei[1]=TENSE\_PAST;//完成

break;

case TENSE\_PRESENT:

zhuweibin->wei[1]=TENSE\_PRESENT;//在做

break;

case TENSE\_FUTURE:

zhuweibin->wei[1]=TENSE\_FUTURE;//要去做

break;

}

switch(beibei)

{

case 0:

zhuweibin->wei[2]=0;//被动句

break;

case 1:

zhuweibin->wei[2]=1;//主动句

break;

}

switch(yiwen\_gantan(temp))

{

case 0:

zhuweibin->bin[2]=0;

break;

case 1:

zhuweibin->bin[2]=1;

break;

case 2:

zhuweibin->bin[2]=2;

break;

case 3:

zhuweibin->bin[3]=3;

break;

}

int num\_he=0;

int num\_huo=0;

int zhubin=ZHUYU;

int fucishu=0;

int adj=0;

for(i=0;arr[i][0]!='\0';i++)

{

if(strcmp(arr[i],"和")==0||strcmp(arr[i],"还有")==0||strstr(arr[i],"及")!=NULL)

{

num\_he++;

if(zhubin==BINYU)

//就是它之前是主语的意思

{

zhubin=ZHUYU;

}

else

{

zhubin=BINYU;

}

continue;

}

else if(strcmp(arr[i],"或")==0||strcmp(arr[i],"或者")==0||strcmp(arr[i],"抑或")==0)

{

num\_he++;

if(zhubin==BINYU)

{

zhubin=ZHUYU;

}

else

{

zhubin=BINYU;

}

continue;

}

else if(qinghua[i]==1||qinghua[i]==5)

{

//目前的标号方式是架构在三层标号法之上的

if(zhubin==ZHUYU)

{

//遇到和，或，那么形容词大搬家

if(strcmp(arr[i-1],"和")==0||strcmp(arr[i-1],"还有")==0||strstr(arr[i-1],"及")!=NULL)

{

for(j=3\*adj-1;j>=0;j--)

{

zhuweibin->zhu[10+num\_he\*3+num\_huo\*3+3+j]=zhuweibin->zhu[10+num\_he\*3+num\_huo\*3+j];

}

}

//这个就正常考虑了

zhuweibin->zhu[7+num\_he\*3+num\_huo\*3]=qinghua[i];

zhuweibin->zhu[8+num\_he\*3+num\_huo\*3]=huake[i];

zhuweibin->zhu[9+num\_he\*3+num\_huo\*3]=wuda[i];

zhubin=BINYU;

/\*

解封需要把3都转成6嗷！前后也要变

zhuweibin->zhu[10+num\_he\*3+num\_huo\*3]=phr->num\_pre[0];

zhuweibin->zhu[11+num\_he\*3+num\_huo\*3]=phr->num\_pre[1];

zhuweibin->zhu[12+num\_he\*3+num\_huo\*3]=phr->num\_pre[2];

//警钟敲烂！这个是新的表示！！！

\*/

}

else if(zhubin==BINYU)

{

//形容词搬家

for(j=3\*adj-1;j>=0;j--)

{

zhuweibin->zhu[10+num\_he\*3+num\_huo\*3+3+j]=zhuweibin->zhu[10+num\_he\*3+num\_huo\*3+j];

}

//正常考虑

zhuweibin->bin[7+num\_he\*3+num\_huo\*3]=qinghua[i];

zhuweibin->bin[8+num\_he\*3+num\_huo\*3]=huake[i];

zhuweibin->bin[9+num\_he\*3+num\_huo\*3]=wuda[i];

zhubin=ZHUYU;

/\*

解封需要把3都转成6嗷！前后也要变

zhuweibin->bin[10+num\_he\*3+num\_huo\*3]=phr->num\_pre[0];

zhuweibin->bin[11+num\_he\*3+num\_huo\*3]=phr->num\_pre[1];

zhuweibin->bin[12+num\_he\*3+num\_huo\*3]=phr->num\_pre[2];

//警钟敲烂！这个是新的表示！！！

\*/

}

}

else if(qinghua[i]==2)

//动词

{

zhuweibin->wei[6]=0;

//到后面还有更改的机会

zhuweibin->wei[7]=qinghua[i];

zhuweibin->wei[8]=huake[i];

zhuweibin->wei[9]=wuda[i];

num\_he=0;

num\_huo=0;

//在动词这里为宾语做准备

}

else if(qinghua[i]==4)

//副词

{

fucishu++;

zhuweibin->wei[3]=1;

zhuweibin->wei[4]=fucishu;

zhuweibin->wei[10+(fucishu-1)\*3]=qinghua[i];

zhuweibin->wei[11+(fucishu-1)\*3]=huake[i];

zhuweibin->wei[12+(fucishu-1)\*3]=wuda[i];

}

else if(qinghua[i]==3)

//形容词

{

if(qinghua[i+1]==1||qinghua[i+1]==5||

strcmp("而",arr[i])==0||strcmp("而且",arr[i])==0||

strstr(arr[i],"或")!=NULL||strcmp("要么",arr[i])==0)

//就是说下一个是主语或者宾语，或者是一些特殊的可能出现的词语

{

if(zhubin==ZHUYU)

{

zhuweibin->zhu[8+adj\*3]=qinghua[i];

zhuweibin->zhu[8+adj\*3]=huake[i];

zhuweibin->zhu[8+adj\*3]=wuda[i];

adj++;

}

else

{

zhuweibin->bin[8+adj\*3]=qinghua[i];

zhuweibin->bin[8+adj\*3]=huake[i];

zhuweibin->bin[8+adj\*3]=wuda[i];

}

}

else

{

if(zhubin==BINYU)

{

zhuweibin->wei[6]=1;

zhuweibin->wei[7]=qinghua[i];

zhuweibin->wei[8]=huake[i];

zhuweibin->wei[9]=wuda[i];

//没错，就是要把没啥用的原动词覆盖掉

}

}

}

}

//最后别忘了释放内存~

/\*for(i=0;i<10;i++)

{

free(arr[i]);

}

free(arr);\*/

}

int yiwen\_gantan(char\* str)

{

//1是是什么，2是感叹，0是正常语气，3是怎么做

if(strstr(str,"什么")!=NULL)

return 1;

if(strstr(str,"啥")!=NULL)

return 1;

if(strstr(str,"怎")!=NULL)

return 3;

if(strstr(str,"咋")!=NULL)

return 3;

if(strstr(str,"如何")!=NULL)

return 3;

if(strstr(str,"啊")!=NULL)

return 2;

if(strstr(str,"呢")!=NULL)

return 2;

if(strstr(str,"能干什么")!=NULL)

return 3;

if(strstr(str,"能干啥")!=NULL)

return 3;

if(strstr(str,"呀")!=NULL)

return 2;

if(strstr(str,"解释")!=NULL)

return 1;

if(strstr(str,"原因")!=NULL)

return 1;

if(strstr(str,"方案")!=NULL)

return 3;

if(strstr(str,"方法")!=NULL)

return 3;

if(strstr(str,"建议")!=NULL)

return 3;

if(strstr(str,"哪")!=NULL)

return 1;

if(strstr(str,"办法")!=NULL)

return 3;

if(strstr(str,"做法")!=NULL)

return 3;

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：logic

\*\*文件目的：实现逻辑链的搭建

\*\*时间复杂度：\Theta (句子数)

\*\*空间复杂度：\Theta (句子数)

\*/

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<alloc.h>

#include<lasthao.h>

#include<find.h>

#include<logic.h>

//算法的基本思路：引入"虚拟词汇"的概念，用虚拟的词汇进行分析

//函数名称：virtual\_word

//函数目的：实现虚拟词汇的架构

int virtual\_word(ZHUWEIBIN\* arr,ZHUWEIBIN\* virtual\_arr)

{

//static ZHUWEIBIN virtual\_arr[200];

int jushu;

for(jushu=0;;jushu++)

{

if(panduanNULL(arr[jushu])==0)

{

break;

}

}

jushu=jushu-1;

int i,j;

for(i=0;i<jushu-1;i++)

{

for(j=0;j<40;j++)

{

virtual\_arr[i].zhu[j]=arr[i].bin[j];

}

}

//留一个virtual\_arr[jushu-1]的主语不变

return jushu;

}

//函数名称：virtual\_logic

//函数目的：实现虚拟词汇逻辑上的架构

void virtual\_logic(ZHUWEIBIN\* arr,ZHUWEIBIN\* virtual\_arr)

{

int jushu=virtual\_word(arr,virtual\_arr);

int i,j;

for(i=0;i<jushu;i++)

{

int pronn=pron(arr[i]);

if(pronn==1)

{

for(j=0;j<40;j++)

{

virtual\_arr[jushu-1].zhu[j]=arr[i].bin[j];

virtual\_arr[jushu-1].wei[j]=arr[i].wei[j];

}

}

else if(pronn==2)

{

for(j=0;j<40;j++)

{

virtual\_arr[jushu-1].zhu[j]=arr[i].bin[j];

virtual\_arr[jushu-1].wei[j]=arr[i].wei[j];

}

//变成被动

virtual\_arr[jushu-1].wei[2]=1-virtual\_arr[jushu-1].wei[2];

}

}

//不能太多的逻辑连续，不然容易造成雪崩效应，2个是极限

int m;

for(m=0;m<2;m++)

{

for(i=0;i<jushu-1;i++)

{

if(zhuweibin\_same(virtual\_arr[jushu-1].zhu,arr[i].zhu))

{

for(j=0;j<40;j++)

{

virtual\_arr[jushu-1].zhu[j]=arr[i].bin[j];

virtual\_arr[jushu-1].wei[j]=arr[i].wei[j];

//记录它谓语干了什么，反正virtual\_arr的谓语没啥用

}

}

else if(zhuweibin\_same(virtual\_arr[jushu-1].zhu,arr[i].bin))

{

for(j=0;j<40;j++)

{

virtual\_arr[jushu-1].zhu[j]=arr[i].zhu[j];

virtual\_arr[jushu-1].wei[j]=arr[i].wei[j];

}

//变成被动

virtual\_arr[jushu-1].wei[2]=1-virtual\_arr[jushu-1].wei[2];

}

}

}

}

//函数名称：zhuweibin\_same

//函数目的：判断ZHUWEIBIN里的元素是否相同

int zhuweibin\_same(int\* arr1,int\* arr2)

{

int i,temp=0;

for(i=0;i<40;i++)

{

if(arr1[i]!=arr2[i])

{

temp=1;

break;

}

}

return temp;

//1不同，0相同

}

//函数名称：pron

//函数目的：判断句子当中是否存在指他性代词

int pron(ZHUWEIBIN arr)

{

if(arr.zhu[6]==5)

{

if(arr.zhu[7]==1)

{

if(arr.zhu[8]>=3&&arr.zhu[8]<=5)

{

return 1;

}

else if(arr.zhu[8]>=8&&arr.zhu[8]<=10)

{

return 1;

}

else if(arr.zhu[8]==16)

{

return 1;

}

}

}

if(arr.bin[6]==5)

{

if(arr.bin[7]==1)

{

if(arr.bin[8]>=3&&arr.bin[8]<=5)

{

return 2;

}

else if(arr.bin[8]>=8&&arr.bin[8]<=10)

{

return 2;

}

else if(arr.bin[8]==16)

{

return 2;

}

}

}

return 0;

//0是没有,1是作为主语,2是作为宾语

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<include.h>

int MouseX=0,MouseY=0,MouseS=0,press;

int MouseX\_memory,MouseY\_memory;

unsigned int far\* buffer;

union REGS our\_AI\_regs;

int mouse\_color[15][22];

int main()

{

int i=PAGE\_START;

int\* page=&i;

start\_SVGA();

//上面是初始化

//line\_drawing(100,100,500,100,WHITE\_,\*page);//test

while(1)

{

//change\_page(page);

//line\_drawing(100,100,500,100,WHITE\_,\*page);//test

//mouseinit(\*page);

//newmouse(&MouseX,&MouseY,&MouseS,\*page);

//delay(20);

page\_start(\*page);

if(\*page==PAGE\_CLOSE)

{

break;

}

cleardevice();//Turbo C的库函数

}

exit(1);//退出界面

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名称：mouse

\*\* 文件目的：搭建鼠标图形

\*\* 时间复杂度：最小

\*\* 空间复杂度：最小

\*/

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<stdio.h>

#include<stdlib.h>

#include<color.h>

#include<draw.h>

#include<mouse.h>

extern int MouseX;

extern int MouseY;

extern int MouseS;

extern int press;

extern int MouseX\_memory;

extern int MouseY\_memory;

extern unsigned int far\* buffer;

extern union REGS our\_AI\_regs;

int flag=0;

extern int mouse\_color[15][22];

//函数名称：mouseinit

//函数目的：初始化鼠标

void mouseinit(int page)

{

int retcode;

int xmin,xmax,ymin,ymax,x\_max=1024,y\_max=768;

int size;

xmin=2;

xmax=x\_max-1;

ymin=8;

ymax=y\_max-2;

our\_AI\_regs.x.ax=0;

int86(51,&our\_AI\_regs,&our\_AI\_regs);

retcode=our\_AI\_regs.x.ax;

our\_AI\_regs.x.ax=7;

our\_AI\_regs.x.cx=xmin;

our\_AI\_regs.x.dx=xmax;

int86(51,&our\_AI\_regs,&our\_AI\_regs);

our\_AI\_regs.x.ax=8;

our\_AI\_regs.x.cx=ymin;

our\_AI\_regs.x.dx=ymax;

int86(51,&our\_AI\_regs,&our\_AI\_regs);

//save\_bk\_mou(0,0);

//mouse\_drawing(MouseX,MouseY,page);

flag=1;

}

//函数名称：mouse\_drawing

//函数目的：画不同形态的鼠标

void mouse\_drawing(int x,int y,int page)

{

switch(MouseS)

{

case 1: //手势鼠标

line\_drawing(x-1,y+9,x-1,y+8,WHITE\_,page,0);

line\_drawing(x,y+7,x,y+11,WHITE\_,page,0);

line\_drawing(x+1,y+6,x+1,y+13,WHITE\_,page,0);

line\_drawing(x+2,y+8,x+2,y+14,WHITE\_,page,0);

line\_drawing(x+3,y-1,x+3,y+15,WHITE\_,page,0);

line\_drawing(x+4,y-2,x+4,y+15,WHITE\_,page,0);

line\_drawing(x+5,y-1,x+5,y+16,WHITE\_,page,0);

line\_drawing(x+6,y+2,x+6,y+16,WHITE\_,page,0);

line\_drawing(x+7,y+3,x+7,y+17,WHITE\_,page,0);

line\_drawing(x+8,y+4,x+8,y+17,WHITE\_,page,0);

line\_drawing(x+9,y+5,x+9,y+16,WHITE\_,page,0);

line\_drawing(x+10,y+6,x+10,y+16,WHITE\_,page,0);

line\_drawing(x+11,y+7,x+11,y+13,WHITE\_,page,0);

line\_drawing(x-1,y+9,x-1,y+8,DARK\_GRAY\_,page,0);

line\_drawing(x-1,y+8,x+1,y+6,DARK\_GRAY\_,page,0);

line\_drawing(x+1,y+6,x+3,y+10,DARK\_GRAY\_,page,0);

line\_drawing(x+3,y+10,x+3,y-1,DARK\_GRAY\_,page,0);

line\_drawing(x+5,y-1,x+5,y+5,DARK\_GRAY\_,page,0);

line\_drawing(x+7,y+3,x+7,y+7,DARK\_GRAY\_,page,0);

line\_drawing(x+9,y+5,x+9,y+9,DARK\_GRAY\_,page,0);

line\_drawing(x+11,y+7,x+11,y+13,DARK\_GRAY\_,page,0);

line\_drawing(x+7,y+17,x+3,y+15,DARK\_GRAY\_,page,0);

line\_drawing(x+3,y+15,x+1,y+13,DARK\_GRAY\_,page,0);

line\_drawing(x+1,y+13,x-1,y+9,DARK\_GRAY\_,page,0);

break;

case 2: //光标

line\_drawing(x+1,y-1,x+9,y-1,DARK\_GRAY\_,page,0);

line\_drawing(x+1,y+15,x+9,y+15,DARK\_GRAY\_,page,0);

line\_drawing(x+5,y-1,x+5,y+15,DARK\_GRAY\_,page,0);

break;

case 3: //十字

line\_drawing(x-1,y+7,x+11,y+7,WHITE\_,page,0);

line\_drawing(x+5,y-1,x+5,y+15,WHITE\_,page,0);

break;

default: //默认鼠标

line\_drawing(x,y,x,y+13,WHITE\_,page,0);

line\_drawing(x+1,y+1,x+1,y+12,WHITE\_,page,0);

line\_drawing(x+2,y+2,x+2,y+11,WHITE\_,page,0);

line\_drawing(x+3,y+3,x+3,y+10,WHITE\_,page,0);

line\_drawing(x+4,y+4,x+4,y+12,WHITE\_,page,0);

line\_drawing(x+5,y+5,x+5,y+9,WHITE\_,page,0);

line\_drawing(x+5,y+11,x+5,y+14,WHITE\_,page,0);

line\_drawing(x+6,y+6,x+6,y+9,WHITE\_,page,0);

line\_drawing(x+6,y+13,x+6,y+15,WHITE\_,page,0);

line\_drawing(x+7,y+7,x+7,y+9,WHITE\_,page,0);

line\_drawing(x+8,y+8,x+8,y+9,WHITE\_,page,0);

line\_drawing(x+9,y+9,x+9,y+9,WHITE\_,page,0);

line\_drawing(x-1,y-1,x-1,y+14,DARK\_GRAY\_,page,0);

line\_drawing(x-1,y+14,x+3,y+11,DARK\_GRAY\_,page,0);

line\_drawing(x+3,y+11,x+3,y+12,DARK\_GRAY\_,page,0);

line\_drawing(x+3,y+12,x+4,y+13,DARK\_GRAY\_,page,0);

line\_drawing(x+4,y+13,x+4,y+14,DARK\_GRAY\_,page,0);

line\_drawing(x+4,y+14,x+7,y+17,DARK\_GRAY\_,page,0);

line\_drawing(x+7,y+17,x+7,y+13,DARK\_GRAY\_,page,0);

line\_drawing(x+7,y+13,x+6,y+12,DARK\_GRAY\_,page,0);

line\_drawing(x+6,y+12,x+6,y+11,DARK\_GRAY\_,page,0);

line\_drawing(x+6,y+11,x+5,y+10,DARK\_GRAY\_,page,0);

line\_drawing(x+5,y+10,x+11,y+10,DARK\_GRAY\_,page,0);

line\_drawing(x+11,y+10,x-1,y-2,DARK\_GRAY\_,page,0);

break;

}

}

//函数名称：mread

//函数目的：读取寄存器信息

void mread(int \*nx,int \*ny,int \*nbuttons)

{

our\_AI\_regs.x.ax=3;

int86(51,&our\_AI\_regs,&our\_AI\_regs);

\*nx=our\_AI\_regs.x.cx;

\*ny=our\_AI\_regs.x.dx;

\*nbuttons=our\_AI\_regs.x.bx;

}

//函数名称：newmouse

//函数目的：更新鼠标状态

void newmouse(int \*nx,int \*ny,int \*nbuttons,int page)

{

int xn,yn,buttonsn;

int x0=\*nx,y0=\*ny,buttons0=\*nbuttons;

mread(&xn,&yn,&buttonsn);

\*nx=xn;

\*ny=yn;

\*nbuttons=buttonsn;

if(buttons0==\*nbuttons)

\*nbuttons=0; //使得能连续按键

if(xn==x0&&yn==y0&&buttonsn==buttons0)

return; //鼠标状态不变则直接返回S

//mouse\_drawing(\*nx,\*ny,page);

//clrmous(MouseX,MouseY,page);

//truly\_clean\_mouse(MouseX,MouseY); //说明鼠标状态发生了改变

//save\_bk\_mou(\*nx,\*ny);

//drawmous(\*nx,\*ny,page);

}

//函数名称：save\_bk\_mou

//函数目的：存鼠标背景

void save\_bk\_mou(int nx,int ny)

{

int size;

size=imagesize(nx-1,ny-2,nx+11,ny+17);

if(buffer!=NULL)

{

free(buffer);

buffer=(unsigned int\*)malloc(size);

}

if(buffer!=NULL)

{

getimage(nx-1,ny-2,nx+11,ny+17,buffer);

}

else

{

printf("Error");

}

}

//函数名称：clrmous

//函数目的：清除鼠标

void clrmous(int nx,int ny,int page)

{

if(flag==1)

{

setwritemode(XOR\_PUT);

mouse\_drawing(nx,ny,page);

putimage(nx-1,ny-2,buffer,COPY\_PUT);

free(buffer);

flag=0;

setwritemode(COPY\_PUT);

}

}

//函数名称：drawmous

//函数目的：画鼠标且能清除

void drawmous(int nx,int ny,int page)

{

if(flag==0)

{

//setwritemode(COPY\_PUT);

mouse\_drawing(nx,ny,page);

flag=1;

}

}

//函数名称：mouse\_press

//函数目的：判断鼠标的点击情况

//如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0

int mouse\_press(int x1, int y1, int x2, int y2)

{

//在框中点击，则返回1

if(MouseX>x1

&&MouseX<x2

&&MouseY>y1

&&MouseY<y2

&&press==1)

{

return 1;

}

//在框中未点击，则返回2

else if(MouseX>x1

&&MouseX<x2

&&MouseY>y1

&&MouseY<y2

&&press==0)

{

return 2;

}

//在框中点击右键，则返回3

else if(MouseX>x1

&&MouseX<x2

&&MouseY>y1

&&MouseY<y2

&&press==2)

{

return 3;

}

else

{

return 0;

}

}

//函数名称：memory\_mouse

//函数目的：记忆鼠标清除的矩形像素点

void memory\_mouse(int x,int y,int page)

//x=MouseX,y=MouseY

{

int i,j;

for(i=0;i<15;i++)

{

for(j=0;j<22;j++)

{

mouse\_color[i][j]=get\_pixel\_color(x-2+i,y-3+j,page);

}

}

MouseX\_memory=x;

MouseY\_memory=y;

}

//函数名称：truly\_clean\_mouse

//函数目的：实现鼠标的清除

void truly\_clean\_mouse(int x,int y,int page)

//x=MouseX,y=MouseY

{

int x0,y0;

for(x0=x-2;x0<=x+12;x0++)

{

for(y0=y-3;y0<=y+18;y0++)

{

point\_drawing(x0,y0,mouse\_color[x0-x+2][y0-y+3],page);

}

}

}

//函数名称：get\_pixel\_color

//函数目的：获取像素点颜色

int get\_pixel\_color(int x,int y,int page)

{

unsigned int far \*const buffer=(unsigned int far\*)0xa0000000L;

if(x<0||x>XMAX||y<0||y>YMAX)

{

return -1;//返回-1表示坐标超出范围

}

// 计算偏移量

unsigned long long int delta=((unsigned long long int)y<<10)+x;

// 计算所在页

int i=y/32;

int eventual\_page=(page>>16)+i;

pageturn(eventual\_page);

// 返回该像素点的颜色

return buffer[delta];

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int atoi\_f(char\* a, int\* num\_arr, int n, int ci\_xing, int\* c);

int atoI(const char\* str);

int get\_num(char\* a);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PHR结构体：用于存储断句后句子中各个成分的标号，以及替代转化之后的标号

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef struct

{

char\* str; // 存储段句后的这个短语

int num\_arr[15]; //存储用小数点分隔后的标号，如果没有找到num\_arr[0]==0

int num\_pre[15]; //存储一旦可以被关键词替代，替代后的词语的标号

int x; //用于判断该词是否需要关键词替代，x=1则可以被替代，x=0则不能被替代

} PHR;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Temp结构体：用于临时存储搜索文件时暂时搜到的标号+短语或词语

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef struct

{

char num[20];

char phrase[100];

} Temp;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：mark

函数目的：给断句后句子中的成分标号

函数参数：str：断句后的短语数组（字符串数组）

ceng：记录标号中有几个数字，如1.23.有2个，是关键词，不可被替代；而1.23.4就有3个 ，不是关键词，需要被替代

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PHR\* mark(char\*\* str, int\* ceng)

{

int ceng\_shu;//用于记录标号中有几个数字，如1.23.有2个；而1.23.4就有3个

//static PHR arr[300]; // 声明结构体

PHR arr[30]={0};

int i = 0, j = 0, k = 0;

for(i=0;i<30;i++)

{

arr[i].str=(char\*)malloc(30);

memset(arr[i].str,0,30);

for(j=0;j<15;j++)

{

arr[i].num\_arr[j]=0;

arr[i].num\_pre[j]=0;

}

arr[i].x=0;

}

i=0;

j=0;

//int m=0;

FILE\* file;

//static Temp temp;

Temp temp;

char c; // 用c来读取file指向的字符

// arr[i]是存断完的句子中的各个部分的信息的结构体

while (str[i])

{

arr[i].str = str[i]; // 先存储初始内容

i++;

}

i = 0; // 清零，以便下次使用

while (str[i])

{

// 重置临时变量

memset(&temp, 0, sizeof(temp));

// 1.查找是不是名词里面的

file = fopen("C:\\CODE\\word\\mingci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\mingci.txt\n");

}

while (!feof(file))// 当名词文件尚未检索完毕时

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中一个一个比较

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

// if (strstr(temp.phrase, str[i]) != NULL) // 在名词对应文件中找到了（在名词里面）

if (strstr(temp.phrase, str[i]) != NULL) // 在文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 1, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 1)

{

i++;

continue;

}

// 2.查找是不是动词里面的

file = fopen("C:\\CODE\\word\\dongci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\dongci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在动词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 2, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 2)

{

i++;

continue;

}

// 3.查找是不是形容词里面的

file = fopen("C:\\CODE\\word\\xingrongci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\xingrongci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中一个一个比较

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在形容词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 3, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 3)

{

i++;

continue;

}

// 4.查找是不是副词里面的

file = fopen("C:\\CODE\\word\\fuci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\fuci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在副词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 4, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 4)

{

i++;

continue;

}

// 5.查找是不是连词里面的

file = fopen("C:\\CODE\\word\\lianci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\lianci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在连词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 5, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 5)

{

i++;

continue;

}

// 6.查找是不是介词里面的

file = fopen("C:\\CODE\\word\\jieci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\jieci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在动词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 6, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 6)

{

i++;

continue;

}

// 7.查找是不是代词里面的

file = fopen("C:\\CODE\\word\\daici.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\daici.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在动词对应文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 7, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 7)

{

i++;

continue;

}

// 8.查找是不是数量词里面的

file = fopen("C:\\CODE\\word\\shuliangci.txt", "rt"); // 以只读模式打开文件,file自动指向开头

if (file == NULL)

{

perror("无法打开文件");

return 0;

}

else

{

//printf("文件成功打开: D:\\shuliangci.txt\n");

}

while (!feof(file))

{

// 避免开始文件指针指向空白区域

do

{

c = fgetc(file); // 文件指针右移一个单位

} while (c != EOF && (c > '9' || c < '0')); // 如果不是标号中的数字

if (c == EOF)

{

break;

}

fseek(file, -1, SEEK\_CUR); // 复原

// 把文件中的%d.%f和词语存到temp结构体中

j = 0; // 清零，以便下次使用

while ((c = fgetc(file)) != EOF && ((c >= '0' && c <= '9') || c == '.')) // 存浮点数

{

temp.num[j++] = c;

}

temp.num[j] = '\0'; // 添加字符串结束符

k = 0; // 清零，以便下次使用

while (c != EOF && c < '0') // 存短语

{

temp.phrase[k++] = c;

c = fgetc(file);

}

temp.phrase[k] = '\0'; // 添加字符串结束符

fseek(file, -1, SEEK\_CUR); // 复原

if (strstr(temp.phrase, str[i]) != NULL) // 在文件中找到了

{

int num\_points = get\_num(temp.num);//计算有几个点

if (atoi\_f(temp.num, arr[i].num\_arr, num\_points + 1, 8, &ceng\_shu) == 1) // 如果这个是最底层的词语，可以进行替代

{

arr[i].x = 1;

}

else

{

arr[i].x = 0;

}

ceng[i] = ceng\_shu;

break;//仅推出这一层循环

}

}

fclose(file); // 关闭文件

if (arr[i].num\_arr[0] == 8)

{

i++;

continue;

}

i++; // 准备给输入内容在断句后的下一个成分进行编号

}

return arr;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：atoi\_f

函数目的：提取文件中已经找到的短语前面对应的数字字符串并将其转化为数组，以便在将标号存入PHR结构体的arr.num\_arr中

函数参数： a：已经找到的短语前面对应的数字字符串

num\_arr：是PHR结构体中arr.num\_arr的形参

n：数字字符串里面有几个'.'

ci\_xing：名词标为1，动词标为2，等等

c：记录标号中有几个数字，也就是层数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int atoi\_f(char\* a, int\* num\_arr, int n, int ci\_xing, int\* c)

{

int i = 0, j = 0, k = 0;

static char s1[5]; // 存储第一个小数点前面的数

static char s2[5]; // 存储第二个小数点前面的数

num\_arr[0] = ci\_xing;

\*c = 2;

// 存储第一个小数点左边的数，表示词性之后的第一级分类

while (a[i] != '.' && a[i] != '\0')

{

s1[j++] = a[i++];

}

s1[j] = '\0'; // 添加字符串结束符

num\_arr[1] = atoI(s1);

for (k = 2; k <= n + 1; k++)

{

if (a[i] == '.' && a[i + 1] <= '9' && a[i + 1] >= '0') // 注意标号中不能有1.0的情况

{

i++; // 把小数点跳过去

j = 0;

while (a[i] != '.' && a[i] != '\0')

{

s2[j++] = a[i++];

}

s2[j] = '\0'; // 添加字符串结束符

num\_arr[k] = atoI(s2);

(\*c)++;

}

}

if (\*c == n)

{

return 2;// 这种情况说明没有可以替代这个词语的，也就是它是分类的名称

}

else

{

return 1;// 这种情况说明这个词语是最底层的词语,需要一个上层的来替代

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：remark

函数目的：对标完号的PHR结构体数组进行加工

函数参数：str：断句后的短语数组（字符串数组）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PHR\* remark(char\*\* str)

{

int i = 0;

int j;

int ceng[300];

//static PHR\* arr;

PHR\* arr;

arr = mark(str, ceng);

//while (str[i])

{

if (arr[i].num\_arr[0] != 0)

{

for (j = 0; j < (ceng[i] - arr[i].x); j++)

{

arr[i].num\_pre[j] = arr[i].num\_arr[j];

}

}

i++;

}

x:

return arr;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：get\_num

函数目的：测量出标号里面有几个点，也就是在词汇树中属于第几层

函数参数：a：断句后的短语对应的数字字符串

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int get\_num(char\* a)

{

int n = 0, i = 0;

while (a[i] != '\0')

{

if (a[i] == '.')

{

n++;

}

i++;

}

return n;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：atoI

函数目的：因为BC库里面没有这个函数，与atoi作用相同

函数参数：str：数字字符串

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int atoI(const char\* str)

{

int result = 0;

int sign = 1;

// 跳过空白字符

while (\*str == ' ' || \*str == '\t')

{

str++;

}

// 处理符号

if (\*str == '-')

{

sign = -1;

str++;

}

else if (\*str == '+')

{

str++;

}

// 逐字符转换

while (\*str >= '0' && \*str <= '9')

{

result = result \* 10 + (\*str - '0');

str++;

}

return sign \* result;

}

PHR\* input(char \*\*word,char\* biaohao)

{

//输入词语

int m;

char shapowanyi[100]={0};

int ceng[8];//分别记录对应几个真实标号

char\* inputArray[3]; // 真正调用的时候可以根据实际情况动态分配内存,现在假设是8个

inputArray[0]=(char\*)malloc(1001);

inputArray[1]=(char\*)malloc(1001);

inputArray[2]=(char\*)malloc(1001);

memset(inputArray[0],0,1001);

memset(inputArray[1],0,1001);

memset(inputArray[2],0,1001);

int i = 0;

int j = 0;

char temp[50];

memset(biaohao,0,100);

//for (i = 0; i < 1; i++)

//{

// printf("请输入第%d个词语（代表断句后的句子成分）: ", i + 1);

// fgets(word[i], 100, stdin);

// word[i][strcspn(word[i], "\n")] = 0; // 去除换行符

//}

for (i = 0; i < 1; i++)

{

inputArray[i] = word[i];

}

//inputArray[2] = NULL;

// 调用remark函数

//static PHR\* phrases = remark(inputArray);

PHR\* phrases = remark(inputArray);

for (i = 0; i < 1; i++)

{

if (phrases[i].num\_arr[0] != 0)

{

if(phrases[i].num\_arr[3]==0)

{

//printf("标号: ");

while (phrases[i].num\_arr[j] != 0)

{

//printf("%d ", phrases[i].num\_arr[j]);

itoa(phrases[i].num\_arr[j],shapowanyi,10);

strcat(biaohao,shapowanyi);

memset(shapowanyi,0,100);

strcat(biaohao,".");

memset(temp,0,50);

j++;

}

}

//printf("\n");

j = 0;

//printf("标号pre: ");

//while (phrases[i].num\_pre[j] != 0)

//{

// //printf("%d ", phrases[i].num\_pre[j]);

// itoa(phrases[i].num\_pre[j],shapowanyi,10);

// strcat(biaohao,shapowanyi);

// strcat(biaohao,".");

// memset(temp,0,50);

// j++;

//}

//printf("\n%d", phrases[i].x);

//itoa(phrases[i].x,shapowanyi,10);

//strcat(biaohao,shapowanyi);

//memset(shapowanyi,0,100);

//strcat(biaohao,".");

//printf("\n\n\n");

}

else

{

//printf("未找到匹配的短语: %s\n\n\n", phrases[i].str);

}

}

x0:

//free(inputArray);

//free(inputArray[0]);

//free(inputArray[1]);

return 0;

}

PHR\* input2(char word[1][101],char\* biaohao)

{

// 输入词语

int m;

char shapowanyi[100]={0};

int ceng[8];//分别记录对应几个真实标号

char\* inputArray[3]; // 真正调用的时候可以根据实际情况动态分配内存,现在假设是8个

inputArray[0]=(char\*)malloc(1001);

inputArray[1]=(char\*)malloc(1001);

inputArray[2]=(char\*)malloc(1001);

memset(inputArray[0],0,1001);

memset(inputArray[1],0,1001);

memset(inputArray[2],0,1001);

int i = 0;

int j = 0;

char temp[50];

memset(biaohao,0,100);

//for (i = 0; i < 1; i++)

//{

// printf("请输入第%d个词语（代表断句后的句子成分）: ", i + 1);

// fgets(word[i], 100, stdin);

// word[i][strcspn(word[i], "\n")] = 0; // 去除换行符

//}

for (i = 0; i < 1; i++)

{

inputArray[i] = word[i];

}

//inputArray[2] = NULL;

// 调用remark函数

//static PHR\* phrases = remark(inputArray);

PHR\* phrases = remark(inputArray);

for (i = 0; i < 1; i++)

{

if (phrases[i].num\_arr[0] != 0)

{

if(phrases[i].num\_arr[3]==0)

{

//printf("标号: ");

while (phrases[i].num\_arr[j] != 0)

{

//printf("%d ", phrases[i].num\_arr[j]);

itoa(phrases[i].num\_arr[j],shapowanyi,10);

strcat(biaohao,shapowanyi);

memset(shapowanyi,0,100);

strcat(biaohao,".");

memset(temp,0,50);

j++;

}

}

//printf("\n");

j = 0;

//printf("标号pre: ");

//while (phrases[i].num\_pre[j] != 0)

//{

// //printf("%d ", phrases[i].num\_pre[j]);

// itoa(phrases[i].num\_pre[j],shapowanyi,10);

// strcat(biaohao,shapowanyi);

// strcat(biaohao,".");

// memset(temp,0,50);

// j++;

//}

//printf("\n%d", phrases[i].x);

//itoa(phrases[i].x,shapowanyi,10);

//strcat(biaohao,shapowanyi);

//memset(shapowanyi,0,100);

//strcat(biaohao,".");

//printf("\n\n\n");

}

else

{

//printf("未找到匹配的短语: %s\n\n\n", phrases[i].str);

}

}

x0:

//free(inputArray);

//free(inputArray[0]);

//free(inputArray[1]);

return 0;

}

/\*

int main()

{

char\*\* arr=(char\*\*)malloc(100\*sizeof(char\*));

//提示一下，第一维度是第几个词，第二维度是这个词的第几个字符

int i,j;

if(arr==NULL)

{

printf("内存分配失败了！");

//delay(2000);

return 0;

}

for(i=0;i<100;i++)

{

arr[i]=(char\*)malloc(500\*sizeof(char));

if(arr[i]==NULL)

{

printf("内存分配失败了！");

return 0;

}

}

for(int i=0;i<1;i++)

{

fgets(arr[i],20,stdin);

}

char biaohao[50]={0};

for(i=0;i<20;i++)

{

printf("%c",biaohao[i]);

}

}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\* 文件名称：page

\*\* 文件目的：实现换页

\*\* 时间复杂度：OH(1)

\*\* 空间复杂度：OH(1)

\*/

#include<stdio.h>

#include<pstat.h>

#include<pregi.h>

#include<page.h>

#include<dos.h>

#include<mouse.h>

#include<pchat.h>

void change\_page(int \*page)//后续还有什么参量往里面加就行，要不就默认别的是void就可以

{

//clrmous(MouseX,MouseY,\*page);

static int initial\_page=PAGE\_START;//起始点开始

int current\_page=\*page;

switch(\*page)

{

case PAGE\_TEST:

\*page=page\_test();

break;

case PAGE\_START:

\*page=page\_start(PAGE\_START);

break;

case PAGE\_CHAT1:

\*page=page\_chat1();

clean\_chat2();

break;

case PAGE\_CHAT2:

\*page=page\_chat2();

clean\_chat1();

break;

case PAGE\_EXIT:

\*page=page\_exit();

break;

case PAGE\_BEFORE:

\*page=initial\_page;

break;

case PAGE\_REGISTER:

\*page=register\_picture(PAGE\_REGISTER);

break;

default:

printf("不知换页到那一页去了！");

}

initial\_page=current\_page;

mouseinit(\*page);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

/\*文件名称：pchat

/\*文件目的：AI聊天界面

/\*时间复杂度：AI时间复杂度

/\*空间复杂度：实际所有像素对应的空间

\*/

#include<stdio.h>

#include<string.h>

#include<input.h>

#include<AI.h>

#include<page.h>

#include<mouse.h>

#include<dos.h>

#include<pchat.h>

#include<color.h>

#include<draw.h>

#include<hz.h>

#include<fenci.h>

#include<ptext.h>

extern int MouseX,MouseY,MouseS,press;

extern void\* buffer;

extern union REGS our\_AI\_regs;

extern int MouseX\_memory,MouseY\_memory;

#include<graphics.h>

#include<draw.h>

#include<stdarg.h>

#include<conio.h>

#include<Bezier.h>

#include<xiezi.h>

#include<color.h>

#include<dos.h>

#include<stddef.h>

#include<mouse.h>

#include<stdlib.h>

#include<car.h>

#include<dog.h>

#include<boy.h>

#include<fish.h>

#include<allo.h>

#include<view.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数目的：重新定位（按照不同的要求）

x,y:该图像原点（放缩的点）所处位置，用于后面画图

即car(int x,int y,float k,int color,int page)里面的x,y

k:放缩倍数

posi:提供定位数据的数组

posi[0]：上

posi[1]：下

posi[2]：左

posi[3]：右

posi[4]：中x

posi[5]：中y

p0:原图中的初始坐标（刚画出来的）

p0[0]：上

p0[1]：下

p0[2]：左

p0[3]：右

p0[4]：中x

p0[5]：中y

xy0:原图中的原点坐标（刚画出来的）

judge[0]： 确定需要根据什么定位（只是定位，不包括放缩）

1：按照 左、上 定位（停靠）

2：按照 右、上 定位（停靠）

3：按照 左、下 定位（停靠）

4：按照 右、下 定位（停靠）

5：按照 中心定位（中心停靠）

judge[1]: 既控制位置，又限制大小，该函数负责限制大小

如果要是上下左右都想限制的话，横纵比很有可能不一样，图形的形状就会出问题，所以特别设立这几种情况

1：按 上、下 控制大小

2：按 左、右 控制大小

140\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void allo(int\* x,int\* y,float\* k,int page,int\* posi,int\* p0,int\* xy0,int\* judge)

{

/\*\*\*\*\*\*

等式方程

posi[0] = k\*(\*y-xy0[1]+p0[0])-\*y\*(k-1) = (p0[0]-xy0[1])\*k+\*y

posi[1] = k\*(\*y-xy0[1]+p0[1])-\*y\*(k-1) = (p0[1]-xy0[1])\*k+\*y

posi[2] = k\*(\*x-xy0[0]+p0[2])-\*x\*(k-1) = (p0[2]-xy0[0])\*k+\*x

posi[3] = k\*(\*x-xy0[0]+p0[3])-\*x\*(k-1) = (p0[3]-xy0[0])\*k+\*x

posi[4] = k\*(\*x-xy0[0]+p0[4])-\*x\*(k-1) = ((posi[2]+posi[3])/2-xy0[0])\*k+\*x

posi[5] = k\*(\*y-xy0[1]+p0[5])-\*y\*(k-1) = ((posi[0]+posi[1])/2-xy0[1])\*k+\*y

\*\*\*\*\*\*/

/\*

解得：

\*y=posi[0]-(p0[0]-xy0[1])\*k;

\*y=posi[1]-(p0[1]-xy0[1])\*k;

\*x=posi[2]-(p0[2]-xy0[0])\*k;

\*x=posi[3]-(p0[3]-xy0[0])\*k;

\*x=posi[4]-((posi[2]+posi[3])/2-xy0[0])\*k;

\*y=posi[5]-((posi[0]+posi[1])/2-xy0[1])\*k;

\*/

//处理放缩的倍数

switch(judge[1])

{

case 1:

\*k=((float)(posi[1]-posi[0]))/((float)(p0[1]-p0[0]));

break;

case 2:

\*k=((float)(posi[3]-posi[2]))/((float)(p0[3]-p0[2]));

break;

default:

\*k=((float)(posi[3]-posi[2]))/((float)(p0[3]-p0[2]));

break;

}

//具体定位图形位置

switch(judge[0])

{

case 1:

\*y=posi[0]-(p0[0]-xy0[1])\*(\*k);

\*x=posi[2]-(p0[2]-xy0[0])\*(\*k);

posi[1]=posi[0]+\*k\*(p0[1]-p0[0]);

posi[3]=posi[2]+\*k\*(p0[3]-p0[2]);

break;

case 2:

\*y=posi[0]-(p0[0]-xy0[1])\*(\*k);

\*x=posi[3]-(p0[3]-xy0[0])\*(\*k);

posi[1]=posi[0]+\*k\*(p0[1]-p0[0]);

posi[2]=posi[3]-\*k\*(p0[3]-p0[2]);

break;

case 3:

\*y=posi[1]-(p0[1]-xy0[1])\*(\*k);

\*x=posi[2]-(p0[2]-xy0[0])\*(\*k);

posi[0]=posi[1]-\*k\*(p0[1]-p0[0]);

posi[3]=posi[2]+\*k\*(p0[3]-p0[2]);

break;

case 4:

\*y=posi[1]-(p0[1]-xy0[1])\*(\*k);

\*x=posi[3]-(p0[3]-xy0[0])\*(\*k);

posi[0]=posi[1]-\*k\*(p0[1]-p0[0]);

posi[2]=posi[3]-\*k\*(p0[3]-p0[2]);

break;

case 5:

\*x=posi[4]-(p0[4]-xy0[0])\*(\*k);

\*y=posi[5]-(p0[5]-xy0[1])\*(\*k);

posi[0]=posi[5]-\*k\*(p0[5]-p0[0]);

posi[1]=posi[5]+\*k\*(p0[1]-p0[5]);

posi[2]=posi[4]-\*k\*(p0[4]-p0[2]);

posi[3]=posi[4]+\*k\*(p0[3]-p0[4]);

break;

default:

\*x=posi[4]-(p0[4]-xy0[0])\*(\*k);

\*y=posi[5]-(p0[5]-xy0[1])\*(\*k);

posi[0]=posi[5]-\*k\*(p0[5]-p0[0]);

posi[1]=posi[5]+\*k\*(p0[1]-p0[5]);

posi[2]=posi[4]-\*k\*(p0[4]-p0[2]);

posi[3]=posi[4]+\*k\*(p0[3]-p0[4]);

break;

}

}

//画小汽车朝左

void put\_picture\_car1(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={304,618};

int p0[6]={362,662,207,683,445,512};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

secure(posi);

allo(&x,&y,&k,page,posi, p0, xy0,judge);

car(x,y,k,color,page,0);

car\_light(x,y,k,color,page,0);

car\_windows(x,y,k,color,page,0);

}

//画小汽车朝右

void put\_picture\_car2(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={304,618};

int p0[6]={362,662,207,683,445,512};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

secure(posi);

allo(&x,&y,&k,page,posi, p0, xy0,judge);

car(x,y,k,color,page,1);

car\_light(x,y,k,color,page,1);

car\_windows(x,y,k,color,page,1);

}

//画小男孩

void put\_picture\_boy(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={524,353};

int p0[6]={350,689,442,585,513,520};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

secure(posi);

allo(&x,&y,&k,page,posi, p0, xy0,judge);

boy(x,y,k,color,page);

}

//画小狗（右）

void put\_picture\_dog1(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={414,245};

int p0[6]={70,470,216,610,413,270};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

dog\_body(x,y,k,color,page,0);

dog\_eyes(x,y,k,color,page,0);

}

//画填充小狗（右）

void put\_picture\_dog1\_fill(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={414,245};

int p0[6]={70,470,216,610,413,270};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

dog\_body(x,y,k,color,page,0);

fill\_1(color,page);

dog\_eyes(x,y,k,color,page,0);

}

//画小狗（左）

void put\_picture\_dog2(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={414,245};

int p0[6]={70,470,216,610,413,270};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

dog\_body(x,y,k,color,page,1);

dog\_eyes(x,y,k,color,page,1);

}

//画填充小狗（左）

void put\_picture\_dog2\_fill(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={414,245};

int p0[6]={70,470,216,610,413,270};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

dog\_body(x,y,k,color,page,1);

fill\_1(color,page);

dog\_eyes(x,y,k,color,page,1);

}

//画鱼(左)

void put\_picture\_fish1(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={497,411};

int p0[6]={411,490,445,635,540,450};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

secure(posi);

allo(&x,&y,&k,page,posi, p0, xy0,judge);

fish(x,y,k,color,page,0);

fish\_head(x,y,k,color+1,page,0);

fish\_eye(x,y,k,color+2,page,0);

}

//画鱼(右)

void put\_picture\_fish2(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={497,411};

int p0[6]={411,490,445,635,540,450};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

secure(posi);

allo(&x,&y,&k,page,posi, p0, xy0,judge);

fish(x,y,k,color,page,1);

fish\_head(x,y,k,color+1,page,1);

fish\_eye(x,y,k,color+1,page,1);

}

//画连绵不断的山（放缩倍数k为1是从最左面连到最右面）

void put\_picture\_mountain(int\* posi,int\* judge,int page,int color)

{ int x=0;

int y=0;

float k=1.0;

int xy0[2]={1,343};

int p0[6]={0,689,1,1024,512,363};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

mountain(x,y,k,color,page);

}

//画水的波纹

void put\_picture\_water(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={1,343};

int p0[6]={653,697,214,620,417,675};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

water(x,y,k,color,page);

}

//画天空中的云

void put\_picture\_cloud(int\* posi,int\* judge,int page,int color)

{

int x=0;

int y=0;

float k=1.0;

int xy0[2]={1,343};

int p0[6]={39,125,186,302,244,164};

allo(&x,&y,&k,page,posi, p0, xy0,judge);

cloud(x,y,k,color,page);

}

//防止图形越界

void secure(int\* posi)

{

if(posi[0]<=105)

{

posi[1]+=(105-posi[0]);

posi[0]=105;

}

if(posi[1]>=489)

{

posi[0]-=(posi[1]-489);

posi[1]=489;

}

if(posi[2]<=135)

{

posi[3]+=(135-posi[2]);

posi[2]=135;

}

if(posi[3]>=925)

{

posi[2]-=(posi[3]-925);

posi[3]=925;

}

}

//函数名称：page\_chat1

//函数目的：实现聊天界面的鼠标和按键配置

int page\_chat1()

{

static char far input[51][421];

int i;

static int input\_counter=1;

int status=0;

int ha,ah;

int j;

int zishu=0;

int sha=0;

for(i=0;i<51;i++)

{

for(j=0;j<421;j++)

{

input[i][j]='\0';

}

}

char inputinput[101]={0};

int page=PAGE\_CHAT1;

int kkk;

int asc2hanzi;

int keyboard=0;

page\_SVGA\_chat1(page,keyboard,input);

//FILE \*fpp;

//mread(&MouseX,&MouseY,&MouseS);

//clrmous(MouseX,MouseY,page);

delay(20);

//save\_bk\_mou(MouseX,MouseY);

mouseinit(page);

//char input[1001];//支持输入500个汉字（2数字=1汉字）

//char \*input=(char\*)malloc(10001);//支持输入5000个汉字（2数字=1汉字）

//char \*eventual=(char\*)malloc(10001);

//int status=0;

char py[100]={0};

char save[3]={0,0,0};

int choose=0;//0是文生文，1是文生图

int counting=0;

int asc2number=0;

char value;

int hanzinumber=0;

int tempp;

int pianyi;

while(1)

{

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX,MouseY,page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX,MouseY,page);

delay(20);

truly\_clean\_mouse(MouseX,MouseY,page);

//汉字输入

if(MouseX>=85&&MouseX<=935&&MouseY>=609&&MouseY<=768)

{

if(mouse\_press(85,650,935,738)==2)

{

//if(status==0)

{

MouseS=2;

status=2;

}

continue;

}

else if(mouse\_press(85,650,935,738)==1)

{

status=2;

MouseS=0;

color\_change\_chat1(2,page);

while(1)

{

huiqu:

if(24\*asc2number+32\*hanzinumber<20\*32)

{

;

}

else

{

while(1)

{

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

//mread(&MouseX,&MouseY,&press);

memory\_mouse(MouseX, MouseY, page);

//drawmous(MouseX,MouseY,page);

mouse\_drawing(MouseX, MouseY, page);

delay(20);

truly\_clean\_mouse(MouseX, MouseY, page);

//退出

if(MouseX>=964&&MouseX<=1024&&MouseY>=0&&MouseY<=60)

{

if(mouse\_press(964,0,1024,60)==2)

{

//if(status==0)

//{

MouseS=1;

status=1;

//}

continue;

}

else if(mouse\_press(964,0,1024,60)==1)

{

for(i=1;i<51;i++)

{

memset(input[i],0,421);

}

input\_counter=1;

color\_change\_chat1(1,page);

status=1;

MouseS=0;

page=PAGE\_START;

change\_page(&page);

}

}

//文生文按键

if(MouseX>=85&&MouseX<=235&&MouseY>=585&&MouseY<=638)

{

if(mouse\_press(85,585,235,638)==2)

{

//if(status==0)

//{

MouseS=1;

status=3;

//}

continue;

}

else if(mouse\_press(85,585,235,638)==1)

{

//wenshengwen:

input\_counter=0;

for(kkk=0;kkk<51;kkk++)

{

memset(input[kkk],0,421);

}

color\_change\_chat1(3,page);

status=3;

MouseS=0;

choose=0;

strcpy(input[0],"您好！请问有什么可以帮助您的吗？");

input\_counter=1;

}

}

//文生图按键

if(MouseX>=315&&MouseX<=465&&MouseY>=585&&MouseY<=638)

{

if(mouse\_press(315,585,465,638)==2)

{

//if(status==0)

//{

MouseS=1;

status=4;

//}

continue;

}

else if(mouse\_press(315,585,465,638)==1)

{

//wenshengtutu:

input\_counter=0;

for(kkk=0;kkk<51;kkk++)

{

memset(input[kkk],0,421);

}

color\_change\_chat1(4,page);

status=4;

MouseS=0;

choose=1;

}

}

//箭头框

if(MouseX>=947&&MouseX<=1011&&MouseY>=662&&MouseY<=726)

{

if(mouse\_press(947,662,1011,726)==2)

{

//if(status==0)

//{

MouseS=1;

status=5;

//}

continue;

}

else if(mouse\_press(947,662,1011,726)==1)

{

fill\_color(85, 650, 935, 738, SNOW\_, page, 0);

color\_change\_chat1(5,page);

status=5;

MouseS=0;

goto zhenhao;

//执行AI功能的入口函数->后端入口

//fenci(input,eventual);

//fill\_color(85,650,935,738,SNOW\_,page);

}

}

//其他位置

else

{

MouseS=0;

status=0;

}

if(kbhit())

{

value=bioskey(0);

if(value=='\b')

{

if(inputinput[strlen(inputinput)-1]>0)

{

fill\_color(85+24\*asc2number+32\*hanzinumber-24,650,85+24\*asc2number+32\*hanzinumber,738,SNOW\_,page,0);

asc2number--;

counting--;

inputinput[strlen(inputinput)-1]='\0';

}

else

{

fill\_color(85+24\*asc2number+32\*hanzinumber-32,650,85+24\*asc2number+32\*hanzinumber,738,SNOW\_,page,0);

hanzinumber--;

counting-=2;

inputinput[strlen(inputinput)-1]='\0';

inputinput[strlen(inputinput)-1]='\0';

}

goto huiqu;

}

}

}

//zishu=0;

//asc2number=0;

//hanzinumber=0;

//fill\_color(85,650,935,738,SNOW\_,page,0);

}

tempp=input\_method(150+24\*asc2number+32\*hanzinumber,609,save,BLACK\_,py,page,&choose,&keyboard,&pianyi,inputinput,counting,input,&input\_counter);

if(save[0]==32||save[0]==13)

{

continue;

}

if(tempp==666)

{

if(save[0]>='0'&&save[0]<='9')

{

;

}

else if(save[0]=='+'||save[0]=='-'||save[0]=='\*'

||save[0]=='/'||save[0]=='='||save[0]==','

||save[0]=='!'||save[0]=='.'||save[0]=='?')

{

;

}

else

{

continue;

}

}

if(save[0]>0)

{

asc2number++;

}

else if(save[0]<0)

{

hanzinumber++;

}

if(tempp==500)

{

goto wenshengwen;

}

if(tempp==2001)

{

goto wenshengtutu;

}

if(tempp==100)//还需加上y的逻辑运算

{

zhenhao:

strcpy(input[input\_counter++],inputinput);

//要不要删掉最后的6个ASC2字符？？？

//display\_text(input,0,page);

if(input[49][0]!='\0')

{

for(ha=0;ha<51;ha++)

{

for(ah=0;ah<421;ah++)

{

input[ha][ah]='\0';

}

}

strcpy(input[0],"已自动为您开启新对话");

strcpy(input[1],inputinput);

input\_counter=2;

keyboard=0;

}

entrance(page,&keyboard,choose,input,&input\_counter);

asc2number=0;

hanzinumber=0;

zishu=0;

MouseS=0;

counting=0;

memset(inputinput,0,101);

}

else if((tempp==4||tempp==3)&&inputinput[0]!='\0')

{

if(inputinput[counting-1]>0)

{

zishu-=24;

asc2number--;

counting--;

inputinput[counting]='\0';

}

else

{

zishu-=32;

hanzinumber--;

counting-=2;

inputinput[counting]='\0';

inputinput[counting+1]='\0';

}

goto x9;

}

else if((tempp==4||tempp==3)&&inputinput[0]=='\0')

{

//number--;

goto x9;

}

else

{

;

}

if(save[0]==0&&save[1]==0)

{

//if(number>0)

//{

//number--;

//}

goto x9;

}

//fpp=fopen("C:\\CODE\\data\\tempchat.txt","rt+");

//fputc(save[0],fpp);

//fputc(save[1],fpp);

//fputc(save[2],fpp);

//fpp--;

if(save[0]>='0'&&save[0]<='9')

{

save[1]=' ';

}

else if(save[0]>='a'&&save[0]<='z')

{

save[0]='\0';

save[1]='\0';

sha=1;

}

else if(save[0]>='A'&&save[0]<='Z')

{

save[0]='\0';

save[1]='\0';

sha=1;

}

if(save[1]==' ')

{

inputinput[counting++]=save[0];

save[1]='\0';

}

else

{

if(save[0]!='\0')

{

inputinput[counting++]=save[0];

}

if(save[1]!='\0')

{

inputinput[counting++]=save[1];

}

}

if(inputinput[0]>0)

{

asc2hanzi=0;

}

else

{

asc2hanzi=1;

}

if(sha==0)

{

if(save[1]==' ')

{

save[1]='\0';

if(asc2hanzi==0)

string\_drawing(save,85+asc2number\*24+hanzinumber\*32-24,650,32,20,12,-8,BLACK\_,320,320,page,0);

else

string\_drawing(save,85+asc2number\*24+hanzinumber\*32-32,650,32,20,12,-8,BLACK\_,320,320,page,0);

//zishu+=24;

save[1]=' ';

}

else

{

if(asc2hanzi==0)

string\_drawing(save,85+asc2number\*24+hanzinumber\*32-24,650,32,20,12,-8,BLACK\_,320,320,page,0);

else

string\_drawing(save,85+asc2number\*24+hanzinumber\*32-32,650,32,20,12,-8,BLACK\_,320,320,page,0);

//zishu+=32;

}

}

else if(sha==1)

{

counting-=2;

sha=0;

//number--;

}

//string\_drawing(input,85,650,32,20,12,-8,BLACK\_,32,32,page);//test

x9:memset(save,0,2);

//save[0]=0;

//save[1]=0;

memset(py,0,100);

}

continue;

}

}

//退出

if(MouseX>=964&&MouseX<=1024&&MouseY>=0&&MouseY<=60)

{

if(mouse\_press(964,0,1024,60)==2)

{

//if(status==0)

//{

MouseS=1;

status=1;

//}

continue;

}

else if(mouse\_press(964,0,1024,60)==1)

{

for(i=1;i<21;i++)

{

memset(input[i],0,421);

}

input\_counter=1;

color\_change\_chat1(1,page);

status=1;

MouseS=0;

page=PAGE\_START;

change\_page(&page);

}

}

//文生文按键

if(MouseX>=85&&MouseX<=235&&MouseY>=585&&MouseY<=638)

{

if(mouse\_press(85,585,235,638)==2)

{

//if(status==0)

//{

MouseS=1;

status=3;

//}

continue;

}

else if(mouse\_press(85,585,235,638)==1)

{

wenshengwen:

input\_counter=0;

for(kkk=0;kkk<21;kkk++)

{

memset(input[kkk],0,421);

}

color\_change\_chat1(3,page);

status=3;

MouseS=0;

choose=0;

strcpy(input[0],"您好！请问有什么可以帮助您的吗？");

input\_counter=1;

}

}

//文生图按键

if(MouseX>=315&&MouseX<=465&&MouseY>=585&&MouseY<=638)

{

if(mouse\_press(315,585,465,638)==2)

{

//if(status==0)

//{

MouseS=1;

status=4;

//}

continue;

}

else if(mouse\_press(315,585,465,638)==1)

{

wenshengtutu:

input\_counter=0;

for(kkk=0;kkk<21;kkk++)

{

memset(input[kkk],0,421);

}

color\_change\_chat1(4,page);

status=4;

MouseS=0;

choose=1;

}

}

//箭头框

if(MouseX>=947&&MouseX<=1011&&MouseY>=662&&MouseY<=726)

{

if(mouse\_press(947,662,1011,726)==2)

{

//if(status==0)

//{

MouseS=1;

status=5;

//}

continue;

}

else if(mouse\_press(947,662,1011,726)==1)

{

color\_change\_chat1(5,page);

status=5;

MouseS=0;

//执行AI功能的入口函数->后端入口

//fenci(input,eventual);

//fill\_color(85,650,935,738,SNOW\_,page);

}

}

//其他位置

else

{

MouseS=0;

status=0;

}

}

return choose;

}

//函数名称：page\_SVGA\_chat1

//函数目的：画出基本的chat1的界面

int page\_SVGA\_chat1(int page,int keyboard,char input[51][421])

{

int pianyi=0;//偏移量初始为0

line\_drawing(0,576,1024,576,BLACK\_,page,0);

//fill\_color(0,0,1024,575,SPRING\_GREEN\_,page);

fill\_color(0,577,1024,764,GAINSBORO\_,page,0);

fill\_color(85,650,935,738,SNOW\_,page,0);//对话框

fill\_color(85,585,235,638,RED\_,page,0);//文生文按键

fill\_color(315,585,465,638,RED\_,page,0);//文生图按键

//画圆圈

circle\_with\_drawing(979,694,32,ROSY\_BROWN\_,page,0);

//画箭头

for(int i=0;i<=2;i++)

{

line\_drawing(979+i,670,979+i,718,BLUE\_,page,0);

line\_drawing(959+i,690,979+i,670,BLUE\_,page,0);

line\_drawing(999+i,690,979+i,670,BLUE\_,page,0);

}

//写汉字

string\_drawing("文生文",111,595,32,20,12,-8,SNOW\_,1000,1000,page,0);

string\_drawing("文生图",341,595,32,20,12,-8,SNOW\_,1000,1000,page,0);

//退出键

fill\_color(964,0,1024,60,RED\_,page,0);

for(i=-2;i<=2;i++)

{

line\_drawing(968+i,8,1020+i,52,WHITE\_,page,0);

line\_drawing(1020+i,8,968+i,52,WHITE\_,page,0);

}

//对话框界面

strcpy(input[0],"您好！请问有什么可以帮助您吗？");

display\_text(input,0,page,keyboard);

}

//函数名称：color\_change\_chat1

//函数目的：实现chat1的颜色变换

void color\_change\_chat1(int i,int page)

{

int j;

//clrmous(MouseX,MouseY,page);

delay(10);

switch(i)

{

//退出

case 1:

fill\_color(964,0,1024,60,DARK\_GRAY\_,page,0);

break;

//输入

case 2:

;

break;

//文生文按键

case 3:

fill\_color(85,585,235,638,DARK\_GRAY\_,page,0);

delay(1000);

fill\_color(85,585,235,638,RED\_,page,0);

string\_drawing("文生文",111,595,32,20,12,-8,SNOW\_,1000,1000,page,0);

break;

//文生图按键

case 4:

fill\_color(315,585,465,638,DARK\_GRAY\_,page,0);

delay(1000);

fill\_color(315,585,465,638,RED\_,page,0);

string\_drawing("文生图",341,595,32,20,12,-8,SNOW\_,1000,1000,page,0);

break;

case 5:

circle\_with\_drawing(979,694,32,DARK\_GRAY\_,page,0);

delay(1000);

circle\_with\_drawing(979,694,32,ROSY\_BROWN\_,page,0);

for(j=0;j<=2;j++)

{

line\_drawing(979+j,670,979+j,718,BLUE\_,page,0);

line\_drawing(959+j,690,979+j,670,BLUE\_,page,0);

line\_drawing(999+j,690,979+j,670,BLUE\_,page,0);

}

break;

}

}

1. **互信息值训练函数**

/\*

\*\* 函数名称：GB2312互信息值训练函数 main

\*\* 程序作用：训练GB2312互信息值数据库，获得不同汉字间的互信息值，从而为后续断句分词打下基础。

\*\* 时间复杂度：字数×(6766+6766)

\*\* 空间复杂度：6766×6767

\*/

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<locale.h>

#define GB\_NUMBER 6763

#define INPUT\_NUMBER 186962\*2+1//比特位个数，就是汉字总数×2+1

void match(double num,char \*array);

int main()

{

int i,j,m;

long long int jishu=0;

//打开文件，配环境

FILE \*fp;

setlocale(LC\_ALL, "zh\_CN.gb2312");

if((fp=fopen("D:\\BC\\DISK\_C\\CODE\\data\\GB2312.txt","rt+"))==NULL)

{

printf("GB2312数据库无法打开！任意键退出");

getchar();

exit(1);

}

FILE \*fp2;

setlocale(LC\_ALL, "zh\_CN.gb2312");

if((fp2=fopen("D:\\BC\\DISK\_C\\CODE\\train\\互信息值训练数据.txt","rt+"))==NULL)

{

printf("目标数据库无法打开！任意键退出");

getchar();

exit(1);

}

//输入训练字符串，采用动态内存分配

char \*arr=(char\*)malloc(1000000\*sizeof(char));

if(arr==NULL)

{

printf("内存分配失败！任意键退出");

getchar();

exit(1);

}

FILE \*fp3;

if((fp3=fopen("D:\\BC\\DISK\_C\\CODE\\data\\tn.txt","rt+"))==NULL)

{

printf("互信息值训练数据无法打开！任意键退出");

getchar();

exit(1);

}

for(i=0;!feof(fp2);i++)

{

arr[i]=fgetc(fp2);

}

//分配堆区内存实现：double count[GB\_NUMBER][GB\_NUMBER+1]={0};

double \*\*count=(double\*\*)malloc(GB\_NUMBER\*sizeof(double\*));

if(count==NULL)

{

printf("内存分配失败！任意键退出");

getchar();

return 1;

}

for(i=0;i<GB\_NUMBER;i++)

{

count[i]=(double\*)malloc((GB\_NUMBER+1)\*sizeof(double));

if(count[i]==NULL)

{

printf("内存分配失败！任意键退出");

// 释放之前分配的内存

for(j=0;j<i;j++)

{

free(count[j]);

}

free(count);

getchar();

return 1;

}

}

for(i=0;i<GB\_NUMBER;i++)

{

for(j=0;j<GB\_NUMBER+1;j++)

{

count[i][j]=0;

}

}

//从上到下，count对应的汉字与原文件汉字排列顺序相同;

//从左到右，count对应的汉字与原文件汉字排列顺序相同，最后一个是总数

//统计汉字出现的个数，以及两个汉字出现的个数

for(i=0;i<strlen(arr);i+=2)

{//每个汉字循环一次，由于GB2312汉字（非ASC2码部分）对应2 bites,所以步长是2

for(j=0;j<GB\_NUMBER\*2;j+=2)

{

fseek(fp,j,0);//在文件里面一个汉字一个汉字的找

char temp\_char=fgetc(fp);

char temp\_char\_after=fgetc(fp);//先固定住文件中汉字对应的二进制表示

if(temp\_char==arr[i]&&temp\_char\_after==arr[i+1])

{

//当汉字一样时，可以记录总数加1并维护

count[j/2][GB\_NUMBER]++;

printf("%d\n",j);//debug

//下面是填写与下个字符关系所用，如果下个字符是\0,那自然没事的

for(m=0;m<GB\_NUMBER\*2;m+=2)

{

fseek(fp,m,0);//照样一个汉字一个汉字的找

char temp\_char\_2=fgetc(fp);

char temp\_char\_after\_2=fgetc(fp);//固定

if(temp\_char\_2==arr[i+2]&&temp\_char\_after\_2==arr[i+3])

{

count[j/2][m/2]++;//给互信息值对应的部分+1

break;

}

}

break;

}

}

}

fclose(fp2);

fclose(fp);

free(arr);

//这一步是算出各个字符对应的出现频率

double sum=0;

for(i=0;i<GB\_NUMBER;i++)

{

sum+=count[i][GB\_NUMBER];

}

//这一步是算出互信息值个数对应的频率，进行替换

for(i=0;i<GB\_NUMBER;i++)

{

for(j=0;j<GB\_NUMBER;j++)

{

if(count[i][GB\_NUMBER]!=0)

{

count[i][j]/=count[i][GB\_NUMBER];

}

else

{

count[i][j]=0;

}

}

}

for(i=0;i<GB\_NUMBER;i++)

{

for(j=0;j<GB\_NUMBER;j++)

{

if(count[j][GB\_NUMBER]!=0)

{

count[i][j]/=count[j][GB\_NUMBER];

}

else

{

count[i][j]=0;

}

}

}

for(i=0;i<GB\_NUMBER;i++)

{

if(sum!=0)

{

count[i][GB\_NUMBER]/=sum;

}

else

{

count[i][GB\_NUMBER]=0;

}

}

count[GB\_NUMBER-1][GB\_NUMBER]=1;

//这一步是用固定的格式打印出所需要的互信息值数据库表格

char array[13];

for(i=0;i<GB\_NUMBER;i++)

{

fputs("{",fp3);

match(count[i][0],array);

fputs(array,fp3);

for(j=1;j<GB\_NUMBER;j++)

{

match(count[i][j],array);

fputs(array,fp3);

}

fputs("}",fp3);

}

return 0;

}

void match(double num,char \*array)

{

int i;

array[0]='0';

array[1]='.';

for(i=1;i<11;i++)

{

num\*=10;

array[i+1]=(int)num+'0';

num-=(int)num;

}

array[12]=',';

array[13]='\0';

}

1. **stceAC.h stce.cpp**

**该文件由于Borland C++ 内存过小，失败了，但是也写在这里，让这个能用在更好的编译器上。**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_stceAC\_h\_

#define \_stceAC\_h\_

#define PAST 0

#define NOW 1

#define FUTURE 2

typedef enum

{

TENSE\_PAST,

//完成

TENSE\_PRESENT,//我们默认为这个时态

//在做

TENSE\_FUTURE,

//要去做

TENSE\_NONE

}Tense;

struct TrieNode

{

struct TrieNode\* children[53];//子节点

//话说一个TrieNode是有128个儿子，NULL的不行，不空才是好儿子(bushi

struct TrieNode\* failure;//失败指针

//这个指向失败后该去往的地方

Tense tense;//当前时态

//时态问题，这个需要次次记录次次调整

int end;//关键词结束标志

//结束标志，这个是末尾才变的

};

struct TrieNode\* create\_node();

void insert\_keyword(struct TrieNode\* root,const char\* word,Tense tense);

Tense detect\_tense(struct TrieNode\* root,char\* input);

struct TrieNode\* init\_automation();

struct TrieNode\* init\_automation\_2();

Tense tense(char\* input,TrieNode\* tree);

Tense already(char\* input,TrieNode\* tree);

int beidong(char\* str);

void recursiveFunction(TrieNode\* node);

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\*\*文件名称：stceAC

\*\*文件目的：时态等相关内容的判断

\*\*空间复杂度：句子标号长度

\*\*时间复杂度：\Theta(1)

\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<alloc.h>

#include<queue.h>

#include<stce.h>

//用AC自动机，速度真的快（别是WA自动机就行bushi）

//函数名称：create\_node

//函数目的：创建新节点

//最终效果：

// 1 -> 2

//基本思路：

// 对于新节点，先上时态，end标注还不是末尾（待标注为末尾），

// failure失败指针有，但是没有添加，然后返回这个节点本身就行了

struct TrieNode\* create\_node()

{

TrieNode\* node=(TrieNode\*)malloc(sizeof(TrieNode));

//test!!!

for(int i=0;i<53;i++)

{

node->children[i]=(TrieNode\*)malloc(sizeof(TrieNode));

}

node->failure=(TrieNode\*)malloc(sizeof(TrieNode));

node->tense=TENSE\_NONE;

node->end=0;

node->failure=NULL;

return node;

}

//函数名称：insert\_keyword

//函数目的：插入关键词，维护优先级

//最终效果：

// 1-> 2 -> 3

// | L---> 4 -> 5

// |

// L---> 6 -> 7

//基本思路：

// 1.词汇从前往后遍历，所以要写循环；

// 2.node往下降一级，如果没有这个"字母"对应的children就女娲造人

// 3.语法继承上一辈的情况：现在的这个孩子的语法大于biaohao语法的enum值（甚至是最大的none）

// 4.考虑end，这回必是末尾

void insert\_keyword(struct TrieNode\* root,const char\* word,Tense tense)

{

TrieNode\* node=root;

int i;

unsigned char temp;

for(i=0;word[i]!='\0';i++)

{

temp=word[i];

if(node->children[temp]==NULL)

{

node->children[temp]=create\_node();

}

node=node->children[temp];//往儿子方向走

}

if(tense<node->tense||node->tense==TENSE\_NONE)

{

node->tense=tense;//继承父辈的语法

}

node->end=0;//是最后了，因为没有nil

}

//函数名称：build\_failure\_links

//函数目的：构建失败指针

//最终效果：

// @

// 1<-> 2 <-> 3

// | L<---> 4 -> 5 \_\_\_\_\_\_\_\ 2

// | /

// L<---> 6 <-> 7 \_\_\_\_\_\_\_\

//基本思路： / 1

// 1.创建队列

// 2.根节点的失败指针指向自己，根节点但凡有个孩子（循环遍历去看，要求不空），failure指向根节点，放入队列,队列就是"待办事项",而且是BFS的层次遍历

// 3.只要队列不空，就接着查找，构建一个这样的循环，然后标记一下队列最前面的是谁，把它移掉（一次性，用完就扔），因为不用再搞第二遍了嗷！┗|｀O′|┛

// 4.找curr的子节点，循环去找，但凡有个孩子就定义它是child，再定义当前节点的失败指针，然后一路往上追溯，fail是fail的失败指针，层层往上追溯

// 追溯到什么时候为止呢？直到fail是root（去root即可）或者fail的children有它对应的第几个字母（那就去那个对应字母了）

// 5.要是fail有children[i]，那就去fail的children就行了，不然就去root

// 6.孩子的时态就是孩子失败指针的时态

// 7.子节点入队，以期这次BFS完了马上就搞下一个BFS

void build\_failure\_links(struct TrieNode\* root)

{

Queue\* q;

q;//test

q->front;//test

q->rear;//test

//创建队列方便后续BFS

root->failure=root;

//根节点的失败指针指向自己

int i;

struct TrieNode\* child;

struct TrieNode\* fail;

for(i=0;i<70;i++)

{

if(root->children[i]!=NULL)

{

root->children[i]->failure=root;//有第一层子节点的话，指向根节点

push(q,root->children[i]);//把第一层子节点放队列里面

//节点放在队列尾部

}

}

//划掉下面的循环可以跑！

//第一层子节点的失败指针指向根节点

//BFS

while(queue\_empty(q)==0)

//判断队列是否为空，队列不为空时循环

{

//队列的队首节点是curr

struct TrieNode\* curr=pop(q);

//获取队首节点（不删版本）并移除队列队首节点

for(i=0;i<128;i++)

{

if(curr->children[i]!=NULL)

{

//子节点

child=curr->children[i];

//当前节点的失败指针

fail=curr->failure;

while(fail!=root&&fail->children[i]==NULL)

//意思就是把这一根扫描满了

{

fail=fail->failure;

//fail是fail的失败指针

}

//要是还有孩子，那就让fail往下走，不然就回root

child->failure=fail->children[i]?fail->children[i]:root;

if(child->failure->tense!=child->tense)

{

child->tense=child->failure->tense;

}

//子节点，入队！等候总司令对队列的进一步命令！

//push(q,child);

}

}

}

}

//函数名称：detect\_tense

//函数目的：自动机搜索

//基本思路：

// 1.现把时态搞成NONE

// 2.从上往下依次遍历，如果不是root或者children没有找到，就往下走，找failure

// 3.如果下面还有的话，curr就是children了，往下再走一步

// 4.然后定义一个once，once是用来看时态的，通过改result搞时态，只要once不回到root，就一直执行下面的这个程序

// 5.如果once是末端点了（end等于1），那么时态是过去时的返回过去时，要不然result就是once的时态（要求是result就是个NONE或者once的时态enum比result小即优先级更大）

// 6.无论如何，once变成once对应的failure，也就是往错误指针走，继续执行，直到走到root为止

// 7.返回once对应的时态

Tense detect\_tense(struct TrieNode\* root,char\* biaohao)

{

Tense result=TENSE\_NONE;//先搞到默认状态再说

struct TrieNode\* curr=root;

int i;

unsigned char temp;

for(i=0;biaohao[i]!='\0';i++)

{

temp=biaohao[i];

while(curr!=root&&curr->children[temp]==NULL)

//就是说是不是根节点的结尾

{

curr=curr->failure;

}

if(curr->children!=NULL)

{

curr=curr->children[temp];

}

struct TrieNode\* once=curr;

while(once!=root)

{

if(once->end==1)

{

if(once->tense==TENSE\_PAST)

{

return TENSE\_PAST;

}

if(once->tense<result||result==TENSE\_NONE)

{

result=once->tense;

}

}

once=once->failure;

}

}

return (result!=TENSE\_NONE)?result:TENSE\_PRESENT;

}

//函数名称：init\_automation

//函数目的：Tense数据的初始化

struct TrieNode\* init\_automation()

{

struct TrieNode\* root=create\_node();

//我们直接把这个给放在代码段而非放在文件，方便形成一个树进行查找

insert\_keyword(root,"曾",TENSE\_PAST);

insert\_keyword(root,"已",TENSE\_PAST);

insert\_keyword(root,"了",TENSE\_PAST);

insert\_keyword(root,"过",TENSE\_PAST);

insert\_keyword(root,"刚",TENSE\_PAST);

insert\_keyword(root,"才",TENSE\_PAST);

insert\_keyword(root,"先",TENSE\_PAST);

insert\_keyword(root,"先前",TENSE\_PAST);

insert\_keyword(root,"之前",TENSE\_PAST);

insert\_keyword(root,"从前",TENSE\_PAST);

insert\_keyword(root,"过去",TENSE\_PAST);

insert\_keyword(root,"上回",TENSE\_PAST);

insert\_keyword(root,"上次",TENSE\_PAST);

insert\_keyword(root,"那时",TENSE\_PAST);

insert\_keyword(root,"当时",TENSE\_PAST);

insert\_keyword(root,"昨天",TENSE\_PAST);

insert\_keyword(root,"前天",TENSE\_PAST);

insert\_keyword(root,"去年",TENSE\_PAST);

insert\_keyword(root,"前年",TENSE\_PAST);

insert\_keyword(root,"上周",TENSE\_PAST);

insert\_keyword(root,"当年",TENSE\_PAST);

insert\_keyword(root,"以前",TENSE\_PAST);

insert\_keyword(root,"之前",TENSE\_PAST);

insert\_keyword(root,"那年",TENSE\_PAST);

insert\_keyword(root,"那天",TENSE\_PAST);

insert\_keyword(root,"当时",TENSE\_PAST);

insert\_keyword(root,"未来",TENSE\_FUTURE);

insert\_keyword(root,"将来",TENSE\_FUTURE);

insert\_keyword(root,"明天",TENSE\_FUTURE);

insert\_keyword(root,"后天",TENSE\_FUTURE);

insert\_keyword(root,"明年",TENSE\_FUTURE);

insert\_keyword(root,"下周",TENSE\_FUTURE);

insert\_keyword(root,"往后",TENSE\_FUTURE);

insert\_keyword(root,"今后",TENSE\_FUTURE);

insert\_keyword(root,"迟早",TENSE\_FUTURE);

insert\_keyword(root,"早晚",TENSE\_FUTURE);

insert\_keyword(root,"远期",TENSE\_FUTURE);

insert\_keyword(root,"远景",TENSE\_FUTURE);

insert\_keyword(root,"此后",TENSE\_FUTURE);

insert\_keyword(root,"往后",TENSE\_FUTURE);

insert\_keyword(root,"以后",TENSE\_FUTURE);

insert\_keyword(root,"展望",TENSE\_FUTURE);

insert\_keyword(root,"畅想",TENSE\_FUTURE);

insert\_keyword(root,"之后",TENSE\_FUTURE);

insert\_keyword(root,"现在",TENSE\_PRESENT);

insert\_keyword(root,"此刻",TENSE\_PRESENT);

insert\_keyword(root,"此时",TENSE\_PRESENT);

insert\_keyword(root,"当下",TENSE\_PRESENT);

insert\_keyword(root,"目前",TENSE\_PRESENT);

insert\_keyword(root,"即刻",TENSE\_PRESENT);

insert\_keyword(root,"随即",TENSE\_PRESENT);

insert\_keyword(root,"此刻",TENSE\_PRESENT);

insert\_keyword(root,"今",TENSE\_PRESENT);

build\_failure\_links(root);

return root;

}

//函数名称：init\_automation\_2

//函数目的：实现已然未然判断的初始化

struct TrieNode\* init\_automation\_2()

//完成就是TENSE\_PAST

//在做就是TENSE\_PRESENT

//要去做就是TENSE\_FUTURE

{

struct TrieNode\* root2=create\_node();

insert\_keyword(root2,"了",TENSE\_PAST);

insert\_keyword(root2,"过",TENSE\_PAST);

insert\_keyword(root2,"已经",TENSE\_PAST);

insert\_keyword(root2,"刚刚",TENSE\_PAST);

insert\_keyword(root2,"来着",TENSE\_PAST);

insert\_keyword(root2,"方",TENSE\_PAST);

insert\_keyword(root2,"毕",TENSE\_PAST);

insert\_keyword(root2,"已",TENSE\_PAST);

insert\_keyword(root2,"曾",TENSE\_PAST);

insert\_keyword(root2,"刚",TENSE\_PAST);

insert\_keyword(root2,"才",TENSE\_PAST);

insert\_keyword(root2,"都",TENSE\_PAST);

insert\_keyword(root2,"罢",TENSE\_PAST);

insert\_keyword(root2,"完",TENSE\_PAST);

insert\_keyword(root2,"成",TENSE\_PAST);

insert\_keyword(root2,"终于",TENSE\_PAST);

insert\_keyword(root2,"终究",TENSE\_PAST);

insert\_keyword(root2,"算是",TENSE\_PAST);

insert\_keyword(root2,"讫",TENSE\_PAST);

insert\_keyword(root2,"之后",TENSE\_PAST);

insert\_keyword(root2,"以来",TENSE\_PAST);

insert\_keyword(root2,"就",TENSE\_PAST);

insert\_keyword(root2,"便",TENSE\_PAST);

insert\_keyword(root2,"马上",TENSE\_FUTURE);

insert\_keyword(root2,"立刻",TENSE\_FUTURE);

insert\_keyword(root2,"即将",TENSE\_FUTURE);

insert\_keyword(root2,"就要",TENSE\_FUTURE);

insert\_keyword(root2,"准备",TENSE\_FUTURE);

insert\_keyword(root2,"打算",TENSE\_FUTURE);

insert\_keyword(root2,"计划",TENSE\_FUTURE);

insert\_keyword(root2,"正欲",TENSE\_FUTURE);

insert\_keyword(root2,"要",TENSE\_FUTURE);

insert\_keyword(root2,"想",TENSE\_FUTURE);

insert\_keyword(root2,"去",TENSE\_FUTURE);

insert\_keyword(root2,"眼看",TENSE\_FUTURE);

insert\_keyword(root2,"行将",TENSE\_FUTURE);

insert\_keyword(root2,"决意",TENSE\_FUTURE);

insert\_keyword(root2,"立意",TENSE\_FUTURE);

insert\_keyword(root2,"着手",TENSE\_FUTURE);

insert\_keyword(root2,"动手",TENSE\_FUTURE);

insert\_keyword(root2,"开手",TENSE\_FUTURE);

insert\_keyword(root2,"欲",TENSE\_FUTURE);

insert\_keyword(root2,"拟",TENSE\_FUTURE);

insert\_keyword(root2,"着",TENSE\_PRESENT);

insert\_keyword(root2,"呢",TENSE\_PRESENT);

insert\_keyword(root2,"正",TENSE\_PRESENT);

insert\_keyword(root2,"现在",TENSE\_PRESENT);

insert\_keyword(root2,"当下",TENSE\_PRESENT);

insert\_keyword(root2,"眼下",TENSE\_PRESENT);

insert\_keyword(root2,"之际",TENSE\_PRESENT);

insert\_keyword(root2,"这会",TENSE\_PRESENT);

insert\_keyword(root2,"这时",TENSE\_PRESENT);

build\_failure\_links(root2);

return root2;

}

//警告！以下函数只能一个句子一个句子的分析！！！

//函数名称：tense

//函数目的：判断时态

Tense tense(char\* biaohao,TrieNode\* tree)

{

/\*

TrieNode\* tree=(TrieNode\*)malloc(1000);

tree=init\_automation();

\*///这一块可以在执行函数里面添加，会快的多

Tense tense\_;

tense\_=detect\_tense(tree,biaohao);

return tense\_;

}

//函数名称：already

//函数目的：判断已然与未然

Tense already(char\* biaohao,TrieNode\* tree)

{

/\*

TrieNode\* tree=(TrieNode\*)malloc(1000);

tree=init\_automation\_2();

\*///这一块也可以在那个执行函数里面添加

Tense tense\_;

tense\_=detect\_tense(tree,biaohao);

return tense\_;

}

//函数名称：beidong

//函数目的：判断是不是被动句

int beidong(char\* str)

{

if(strstr(str,"被")!=NULL)

{

if(strstr(str,"被子")==NULL)

{

return 1;

//1对应被动句

}

}

return 0;

//0对应主动句

}

void recursiveFunction(TrieNode\* node)

{

if (node == NULL)

{

// 空节点作为一种可能的退出条件

return;

}

// 其他递归逻辑

// 例如遍历子节点

for (int i = 0; i < 256; i++)

{

if (node->children[i] != NULL)

{

recursiveFunction(node->children[i]);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. **互信息值训练数据**

屏幕截图 2025-04-23 152051

1. **课设感想**

**·孟宪喆课设感想**

尽管已拥有一学期的C语言学习基础，但当我将生成式AI作为C课设主题时，才真正体会到从理论到实践的巨大跨越。这不仅是对编程能力的考验，更是一场充满未知与挑战的探索之旅。

由于生成式AI涉及的算法原理、数据处理等内容在以往课程中鲜有涉及，课设初期，我如同置身迷雾之中，在光电大楼的会议室505-1房间，常常对着屏幕上的代码片段反复推敲，一边啃着晦涩的技术文档，一边笨拙地调试软件环境，进度缓慢得让人焦虑。程序设计阶段，困难接踵而至。生成式AI需要构建复杂的模型架构，从数据的预处理到模型的训练与优化，每一步都充满荆棘。起初设想的简单文本生成模型，在实际操作中发现远远无法满足课设要求。为了实现更丰富的功能，我尝试引入深度学习相关算法，但C语言并非深度学习的主流开发语言，没有现成的库可以直接调用，我只能查阅大量资料，自行编写基础算法函数，去实现模拟AI的功能。在最艰难的时刻，我甚至连续一周在光电大楼刷夜，累了就趴在桌子上小憩片刻，醒来继续投入到代码的海洋中。

幸运的是，在这场孤军奋战中，我并非完全孤立无援。老师耐心的指导为我指明方向，学长学姐分享的经验技巧让我少走许多弯路，同学们之间的互相交流也给了我新的灵感。在他们的帮助下，我逐渐攻克难题。我调整了模型架构，简化部分复杂功能，将重点放在核心算法的实现上，让程序既满足课设要求，又能展现生成式AI的魅力。 小组合作在这次课设中同样至关重要。虽然我的项目更偏向个人独立开发，但在与队友交流的过程中，队友从不同的视角提出建议，帮助我发现程序中存在的潜在问题。我们互相检查代码，分享调试经验，在思维的碰撞中不断完善各自的作品。每当我在代码中陷入困境，队友的鼓励和支持都成为我坚持下去的动力。 经过无数个日夜的努力，当看到自己的生成式AI程序能够顺利运行，输出较为理想的结果时，所有的疲惫与辛苦都化作了满满的成就感。

这次课设不仅让我深入掌握了C语言在实际项目中的应用技巧，更锻炼了我的自学能力和解决复杂问题的能力。在光电大楼刷夜的日子，虽然辛苦，但那些与代码相伴的星光，那些攻克难题后的喜悦，都成为我大学时光中最珍贵的回忆。 感谢在课设过程中给予我帮助的每一个人，是你们的陪伴与支持，让我能够在代码的世界里不断探索，实现自我突破，完成这场充满挑战与惊喜的C语言课设之旅。

**·徐多泽课设感想**

本次C语言课程设计的完成，对我来说既是一段充实的经历，也是一种能力的积累。从寒假初步接触生成式AI模拟的项目，到学期中不断完善功能、修复问题，整个过程虽然困难重重，但也让我收获了许多实际经验，提升了自己对编程的理解和掌握。

项目开始时，我和队友一起讨论方案，制定任务分工。由于项目涉及多个模块和功能，前期我花了不少时间去查阅资料、学习图形绘制和内存管理的相关知识。在实际动手编程过程中，很多课上只是听懂的知识点，在真正实现时才发现理解还不够深入，需要反复推敲和实践。这个过程虽然辛苦，但也促使我更加主动地学习与思考。

随着程序的逐步搭建，内存使用问题开始显现，频繁出现闪退、卡顿等现象，因为人工智能模型需要大量的运算和存储，也需要调用许多文件。我和队友一起尝试对程序结构进行优化，合并变量的使用范围，减少数组数量，并使用动态内存分配并及时free等方式控制内存开销。虽然过程繁琐，调试时间也很长，但最终效果稳定，达到了预期目标。

在这个过程中，我也遇到了许多困难。有时为了解决一个小问题，需要查阅大量资料并进行多次尝试；有时候写了很久却发现逻辑思路错误，不得不推倒重来。但每当一个功能成功实现，或者一个bug被修复时，成就感也随之而来，让我更加坚定地投入到项目中。

这次课设不仅锻炼了我的编程能力，也提升了我的团队协作意识。我们在开发过程中互相帮助，交换思路，一起完成调试与修改，不仅提高了效率，也加深了彼此的理解。这种合作精神，对我今后的学习和科研都具有积极意义。

总体来看，课程设计让我把课本知识真正运用到实践中，也让我意识到了理论与实践之间的差距。更重要的是，它激发了我继续深入学习编程的兴趣以及对人工智能大模型的深入理解。我会继续努力，补齐不足，争取在今后的学习中进一步提升自己的综合能力。

1. **课设分工**

|  |  |  |  |
| --- | --- | --- | --- |
| 队员 | 任务 | 代码量 | 文件名 |
| 孟宪喆 | 架构A院第一个SVGA 1:5:5:5的65536色界面 | 898 | draw.cpp  English.cpp  mouse.cpp  xiezi.cpp |
| Bezier曲线  椭圆绘制 | 110 | Bezier.cpp |
| 计算请求语句的预处理 | 641 | 42km.cpp  detect.cpp |
| 高精度计算与数论计算 | 349 | Jisuan.cpp |
| 登录注册、聊天界面（算法） | 1435 | pstat.cpp  pregi.cpp  input.cpp  pchat.cpp |
| A院第一个SVGA输入法 | 426 | hz.cpp |
| 换页函数 | 42 | page.cpp |
| 输入框绘制  历史记录查找与自动清除 | 213 | ptext.cpp |
| 互信息值训练函数  搜集数据、数据清洗、数据训练 | 203 | hxxz.cpp |
| AI功能集总函数 | 220 | AI.cpp |
| 将一句话自动分成若干词语 | 193 | fenci.cpp |
| 排除编码字符数的影响，实现前后两步的衔接 | 67 | trans12.cpp |
| 最终抽象机器语言的生成 | 618 | stce.cpp  lasthao.cpp |
| 通过机器语言给出对应自然语言的回复 | 1218 | find.cpp |
| 特殊问题的直接回答 | 75 | teshu.cpp |
| main函数 | 23 | main.cpp |
| 三大协议栈 | 无 | data文件夹 |
| **合计** | **6731** | |
| 徐多泽 | 对分词后的字符串进行初步标号 | 669 | num.cpp |
| A院首个曲线内填充函数 | 293 | Bezier.cpp |
| 登录、聊天界面（画图） | 150 | pstat.cpp  pchat.cpp |
| 文生图的图像绘制及对每个用贝塞尔曲线描出来的图形的平移、放缩的数据准备（便于后面文生图时的图形平移定位和放缩） | 1539（画图函数较多，按实际代码除以2计算） | view.cpp  boy.cpp  car.cpp  dog.cpp  fish.cpp |
| 画图特殊处理、记忆功能与语义识别函数 | 241 | allo.cpp |
| 文生图的衔接集总函数 | 1242 | wentu.cpp |
| inf语料库 | 无 | inf文件夹 |
| word语料库 | 无 | word文件夹 |
| **合计** | **4134** | |
| **合计** | **10865** | | |