**華中農業大學**

**数据结构实验论文**

**简单的行编辑器的设计**

**Simple line editor design**

**作 者 姓 名：马家乐 导 师：汪毅**

**学 院：理学院 班 级：信息1903**

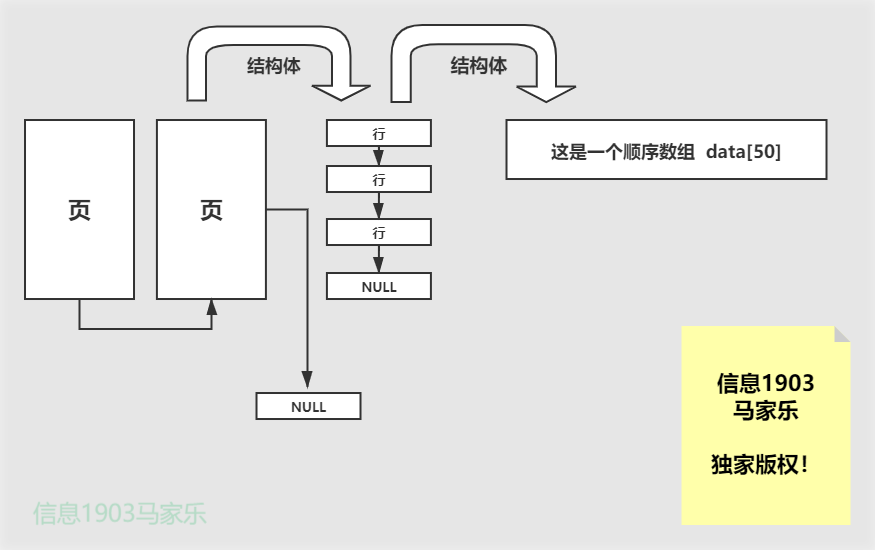
**学 科：信息与计算科学**

**2020年12月**

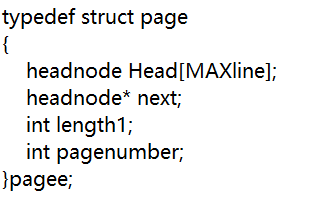
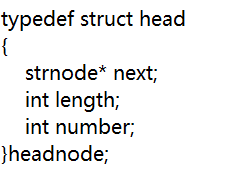
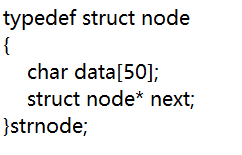
**\*\*\*相关理论基础及函数成果展示\*\*\***

1. **创建基层**

因为行编译器涉及页，行，字符，因此我们需要定义三个结构体，依次嵌套。如下图所示：



解释图意：举例：此时我们拥有两个页，每个页中拥有三个行，每个行中可输入50个字符。通过这样，我们只需给页和行分配空间，再进行类似于单链表的插入操作。再运用pagee类型数组可进行操作。



1. **进行操作**

**（一）.行编译器中包含如下操作：**

1.void Init(); //初始化编辑器

2.void input(); //对指定的行进行输入,#号结束

3.void Delline(); //删除指定的行

4.void List(); //显示文本编辑器的内容

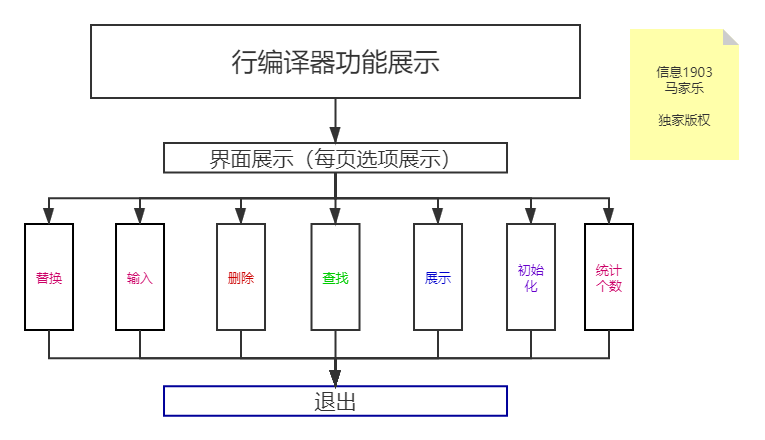
5.int Menu(); //列出选择菜单并进行选择

6.void HeadWord() //美化界面

7.void Find() //查找数据所在位置，并进行统计

8.void jiemian() //设定最开始界面设置

9.void tihuan() //进行行的替换



1. **.深入分析每个函数**
2. **界面展示操作：**先调用jiemian()函数，进行程序开始前的界面设置。

为了使界面更加友好，此时我们用三个printf介绍此应用的名称：《行编辑器》，并欢迎用户，并且著名开发者：信息1903马家乐。

接着利用system(“cls”)，输入回车，进行清屏，美化界面，方便使用者输入操作字符串。

\*下面展示运行结果\*：



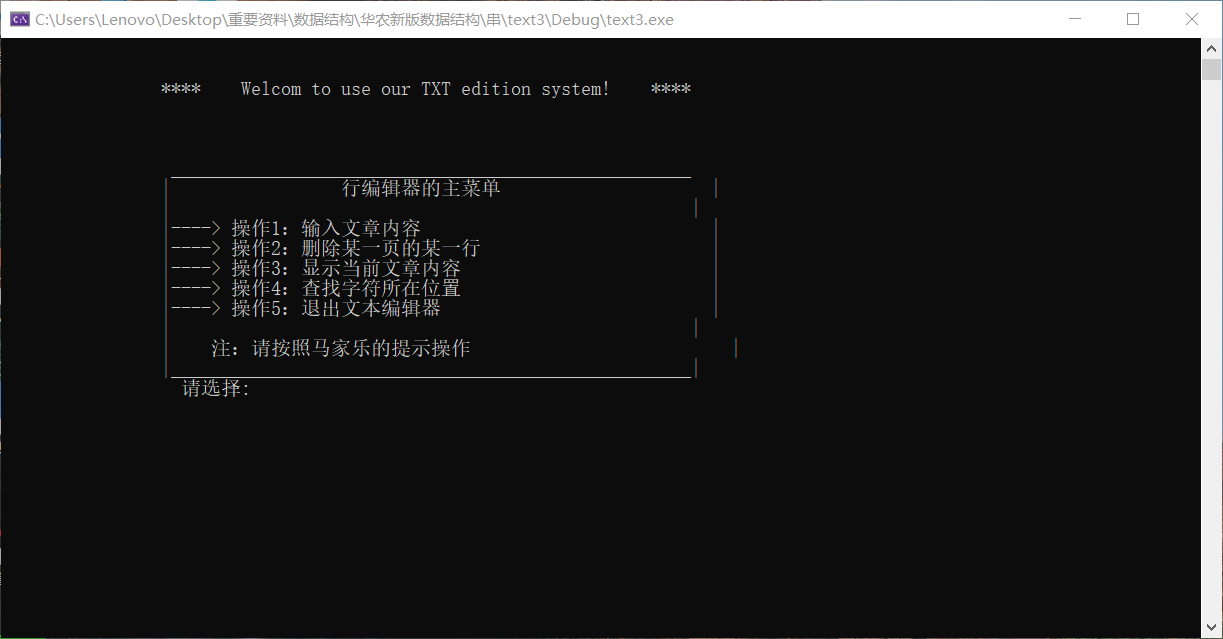
1. **美化界面操作：**再调用Menu()函数和HeadWord()函数，可以使每一次操作时均显示菜单说明。

HeadWord()函数：可以使得每次出现主菜单时候出现页眉，欢迎来到编辑器。

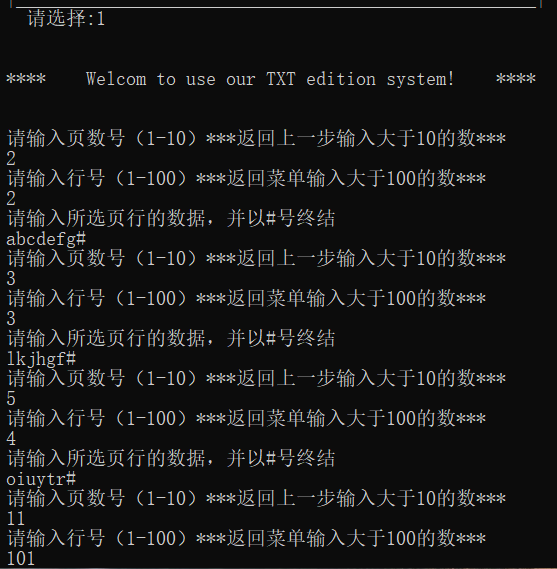
Menu()函数：此时菜单函数提示你选择想进行的操作，进行输入选项，按照马家乐的提示进行操作。

1. 输入操作 2.删除操作 3.显示操作 4.查找操作

5.退出操作6.替代操作 7.统计操作 再进行switch case操作进行筛选。



1. **初始化操作：**利用双重循环，三重结构体从k[ ]中的Head[ ]中的length，并把length设为原始值0，即完成初始化操作，最内核步骤。
2. **输入操作：**需精确到页和行，分配空间进行链式存储。先利用scanf进行输入行数和页数，再对所对应的页和行的字符串进行操作，根据此前结构体的设计，页和行都是链式结构，所以先分配空间，并把输入的值赋予。接着针对所选项进行输入，此时必须有终结点，所以当输入的ch等于#时，暂停输入，输入内部利用单链表的输入，并且由于程序的健壮性，我们设计当超出堆栈范围的时候，利用j%500进行解决，并进入下一行，与此同时length也随之增加。

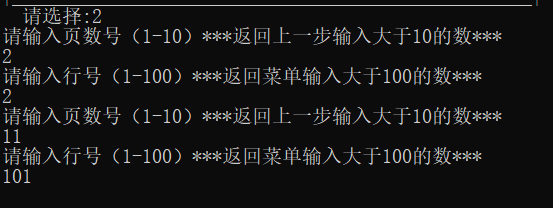


1. **删除操作：**

选择所要删除的页和行，进行删除，注意空间free释放。

先利用scanf进行输入行数和页数，再对所对应的页和行的字符串进行操作，由于程序健壮性，当删除的页行超出范围，即return。

如若符合标准，再判断length是否大于0，若成立，即应用单链表的操作，再应用free()函数，并初始化length和number为0



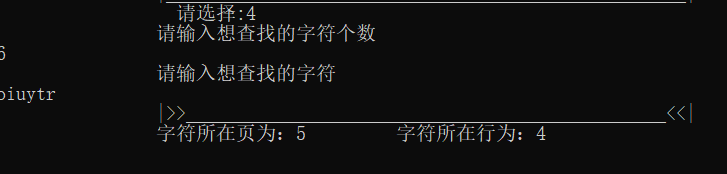
1. **查找操作&统计操作：**

输入字符串长度及字符，利用三重循环进行查找。注意： 分配空间问题。

首先输入字符的个数和字符，注入一个新的数组。

利用三重循环，进行length判断并且利用数组进行每一个字符的对比，如果都相同则输出此时的所在行与所在列。

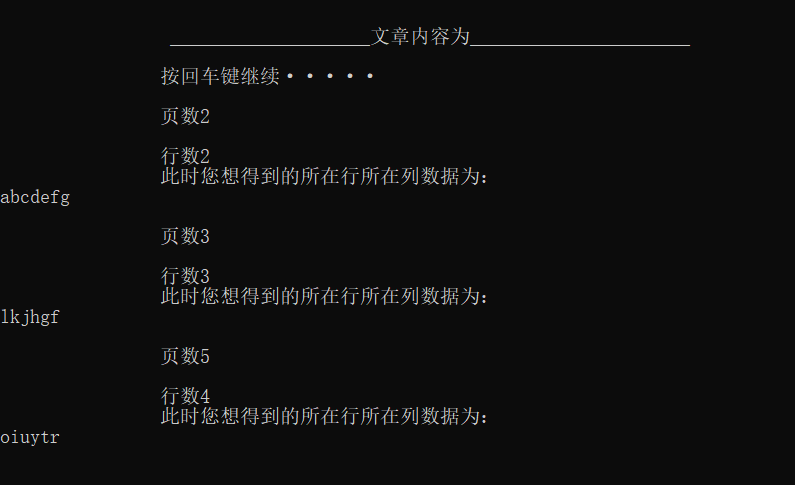
并且在进行查找操作的时候，初始化一个int类型的变量，并把它初始化为0，在进行循环时候把count进行++操作，依次来进行统计个数操作，同时健全了的程序。



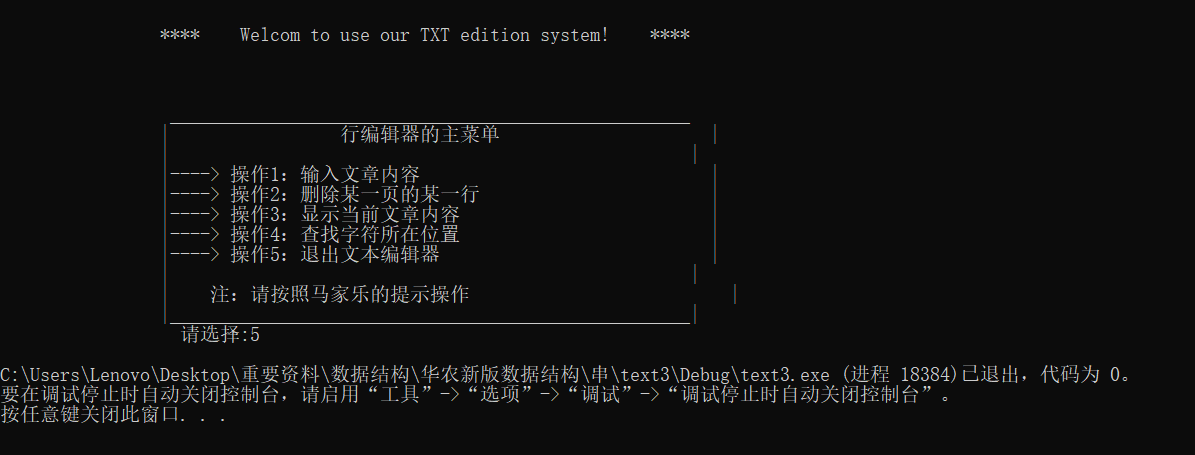
1. **展示操作&替换操作：**

此操作无需对空间进行操作，即利用双重循环遍历到每一行每一页的每一个字符。进而printf输出所在位置的所在数。

此时找到所在行所在页利用新的数组把原字符进行替换。



1. **退出操作：**退出程序。



**\*\*\*代码实现\*\*\***

**一．代码展示**

#include <stdio.h>

#include <stdlib.h>

#define MAXline 100

#define MAXpage 50

typedef struct node

{

char data[50];

struct node\* next;

}strnode;

typedef struct head

{

strnode\* next;

int length;

int number;

}headnode;

typedef struct page

{

headnode Head[MAXline];

headnode\* next;

int length1;

int pagenumber;

}pagee;

pagee k[MAXpage];

void Init()

{

int i, j;

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

{

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

{

k[i].Head[j].length = 0;

}

}

}

void HeadWord()

{

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\n\n");

printf("\t\t\*\*\*\* Welcom to use our TXT edition system! \*\*\*\*\n");

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\n\n");

}

void jiemian()

{

printf("\n\n\t\t\t\t\t\t\t《行编辑器》");

printf("\n\n\t\t\t\t\t\t\t欢迎您使用！");

printf("\n\n\t\t\t\t\t\t开发者：信息1903马家乐！\n");

printf("\n\n\n\n\n\n\n\n\npress Enter to continue...");

getchar();

system("cls");

}

int Menu()

{

int t;

HeadWord();

printf("\n");

printf("\t\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t| 行编辑器的主菜单 |\n");

printf("\t\t| |\n");

printf("\t\t|----> 操作1：输入文章内容 |\n");

printf("\t\t|----> 操作2：删除某一页的某一行 |\n");

printf("\t\t|----> 操作3：显示当前文章内容 |\n");

printf("\t\t|----> 操作4：查找字符所在位置 |\n");

printf("\t\t|----> 操作5：退出文本编辑器 |\n");

printf("\t\t| |\n");

printf("\t\t| 注：请按照马家乐的提示操作 |\n");

printf("\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

printf(" \t\t 请选择:");

scanf("%d", &t);

while (t > 5 || t <= 0)

{

printf("您输入错误，请重新输入\n");

scanf("%d", &t);

}

return t;

}

void input()

{

strnode\* p;

headnode\* t;

int i, a, j, m, LineNum, PageNum;

char ch;

while (1)

{

j = -1;

a = 1;

printf("\t\t请输入页数号（1-10）\*\*\*返回上一步输入大于10的数\*\*\*\n");

printf("\t\t");

scanf("%d", &PageNum);

if (PageNum < 0 || PageNum >= MAXpage)

{

return;

}

printf("\t\t请输入行号（1-100）\*\*\*返回菜单输入大于100的数\*\*\*\n");

printf("\t\t");

scanf("%d", &LineNum);

if (LineNum < 0 || LineNum >= MAXline)

{

return;

}

printf("\t\t请输入所选页行的数据，并以#号终结\n");

i = LineNum;

m = PageNum;

k[m].pagenumber = PageNum;

k[m].next = (strnode\*)malloc(sizeof(strnode));

k[m].Head[i].number = LineNum;

k[m].Head[i].next = (headnode\*)malloc(sizeof(headnode));

p = k[m].Head[i].next;

p->next = NULL;

t = k[m].next;

t->next = NULL;

printf("\t\t");

ch = getchar();

while (ch != '#')

{

j++;

if (j >= 500)

{

p->next = (strnode\*)malloc(sizeof(strnode));

p->next->next = NULL;

p = p->next;

}

p->data[j % 500] = ch;

ch = getchar();

}

k[m].Head[i].length = j + 1;

}

}

void Find()

{

int x, i, m, n, e, o1, o2;

char y, q[100];

int flag = 1;

printf("\t\t请输入想查找的字符个数\t\n");

scanf("%d", &x);

printf("\t\t请输入想查找的字符\t\n");

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

{

scanf("%c", &y);

q[i] = y;

}

printf("\t\t|>>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<<|\n");

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

{

for (n = 0; n <= MAXline; n++)

{

for (e = 0; e < x; e++)

{

if (k[m].Head[n].length>0 && k[m].Head[n].next->data[e] == q[e])

{

flag = 0;

o1 = m;

o2 = n;

}

}

}

}

if (flag == 0)

{

printf("\t\t字符所在页为：");

printf("%d", o1);

printf("\t\t字符所在行为：");

printf("%d", o2);

}

}

void Delline()

{

strnode\* p, \* q;

headnode\* t, \* r;

int i, j,a=0, LineNum, PageNum;

while (1)

{

printf("\t\t请输入页数号（1-10）\*\*\*返回上一步输入大于10的数\*\*\*\n");

printf("\t\t");

scanf("%d", &PageNum);

if (PageNum < 0 || PageNum >= MAXpage)

{

return;

}

printf("\t\t请输入行号（1-100）\*\*\*返回菜单输入大于100的数\*\*\*\n");

printf("\t\t");

scanf("%d", &LineNum);

if (LineNum < 0 || LineNum >= MAXline)

{

return;

}

i = LineNum;

j = PageNum;

p = k[j].Head[i].next;

if (k[j].Head[i].length > 0)

{

while (p != NULL && a<1)

{

q = p->next;

free(p);

p = q;

a++;

}

k[j].Head[i].length = 0;

k[j].Head[i].number = 0;

}

}

}

void List()

{

strnode\* p;

int i, j, m, n, e;

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

{

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

{

if (k[j].Head[i].length > 0)

{

printf("\n");

printf("\n\t\t页数%d", k[j].pagenumber);

printf("\n");

printf("\n\t\t行数%d", k[j].Head[i].number);

printf("\n");

n = k[j].Head[i].length;

m = 1;

p = k[j].Head[i].next;

printf("\t\t此时您想得到的所在行所在列数据为：");

for (e = 0; e < n; ++e)

{

if (e >= 50 \* m)

{

p = p->next;

++m;

}

else

{

printf("%c", p->data[e % 50]);

}

}

}

}

}

printf("\n");

}

int main()

{

int sel;

jiemian();

Init();

while (1)

{

sel = Menu();

switch (sel)

{

case 1:

HeadWord();

input();

break;

case 2:

Delline();

break;

case 3:

HeadWord();

printf("\n\t\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_文章内容为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n");

getchar();

List();

break;

case 4:

Find();

break;

case 5:

exit(0);

}

}

return 0;

}