模式匹配及压缩存储

**一、实验目的**

1.掌握串的模式匹配算法（BF、KMP）

2.掌握特殊矩阵的压缩存储

**二、实验内容**

1.输入两个任意字符串，指定查找位置，统计子串在主串中第一次出现的位置（从0开始）。

随后修改为统计子串出现的次数。

2.将对称矩阵压缩存储，并输出完整矩阵

**三、代码框架**

#include<stdio.h>

#include<string.h>

//主串：MainStr，子串：SubStr，开始查找位置：StartPos

int BF(char \*MainStr, char \*SubStr, int StartPos)

{

}

void main()

{

int pos;

char \*MainStr=new char[50],\*SubStr=new char[30];

printf("请输入主串：");

//补充

printf("请输入子串：");

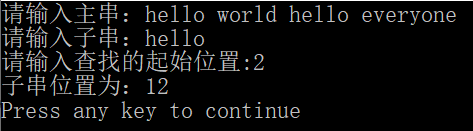
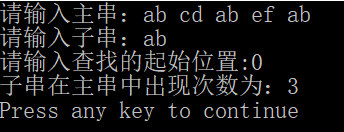
//补充

printf("请输入查找的起始位置");

scanf("%d",&pos);

printf("子串位置为：%d\n",BF(MainStr,SubStr,pos));

}



#include<stdio.h>

#define N 3

int i,j;//全局变量

//向程序中输入对称矩阵

void inputmatrix(int \*a)

{

for(//补充)

for(//补充)

{

scanf("%d",//补充);

}

}

//输出对称矩阵

void outputMatrix(int \*a)

{

for(//补充)

{

for(//补充)

{

if(i>=j)

printf("%d ",//补充);

else

printf("%d ",//补充);

}

printf("\n");

}

}

void main()

{

int \*a=new int[N\*(N+1)/2];

printf("输入%d阶对称矩阵的%d个下三角元素：\n",N,N\*(N+1)/2);

inputmatrix(a,3);

printf("对称矩阵为：\n");

outputMatrix(a,3);

}

