**计算机原理与汇编语言**

**课程设计**

**题目名称：查找字符串中的指定字符**

**姓名：谭哲文**

**学号：8202191123**

**专业：计算机科学与技术**

**班级：计科2105**

**指导教师：贺建飚**

**编写日期：2024.5.12**

**目录**

[1、问题描述： 3](#_Toc2373)

[2、系统设计： 3](#_Toc25258)

[3、源代码清单： 3](#_Toc25983)

[（1）数据段 3](#_Toc7763)

[（2）main函数 4](#_Toc14613)

[（3）print\_str函数 5](#_Toc15748)

[（4）input函数 5](#_Toc1224)

[（5）find\_str函数 5](#_Toc31032)

[（6）turn\_decimal函数 6](#_Toc9570)

[（7）change\_color函数 7](#_Toc3044)

[（8）print\_color\_char函数 7](#_Toc16657)

[4、运行结果测试与分析： 9](#_Toc10846)

[5、结论与心得： 10](#_Toc19244)

# 1、问题描述：

**查找字符串中的指定字符**

基本功能：

1 做一个操作界面，提示操作：输入一串字符串、输入所查找的字符或字符串等；

2 显示出查找到的数目；

3 用不同颜色或闪烁标示出所找到的字符或字符串；

# 2、系统设计：

通过调用DOS 0AH号中断接收键盘的输入，将母串和被查找的子串分别储存到string1和string2中，并且将两个字符串的长度存入len1和len2中。查找依靠两层循环，外层循环从string1的第一个字符枚举查找起点，内层循环比较字符，母串从该起点起，子串从头起，依次向后逐一对比。如果出现不匹配则内层循环退出，外层循环进入下一轮；如果匹配了完整的子串，则说明在母串中成功找到了一个子串，在position中记录此次查找的起点，decimal记录查找到的个数，每次查找成功就+1。下一次的查找起点需跳过子串长度个字符。直至从所有查找起点均查找过，外层循环退出。查找结束后，position数组中记录了所有子串在母串中的起始位置，decimal中记录了查找到的字串数目。再通过二进制转换输出decimal，用变颜色函数将字符串重新输出。

# 源代码清单：

（1）数据段，为之后的查找做准备

stack segment

dw 16 dup(?)

stack ends

data segment

;提示语

message1 db 'Please input a string:',13,10,'$'

message2 db 'Please input a param string:',13,10,'$'

message3 db 'Total number:',13,10,'$'

message4 db 'Result:',13,10,'$'

;接收原字符串

maxlen1 db 100

len1 db ?

string1 db 100 dup('$')

;接收需要查找的字符串

maxlen2 db 50

len2 db ?

string2 db 50 dup('$')

position dw 50 dup(0) ;记录子串开始位置

;查找到的总个数

decimal dw 0

;颜色设置

color\_char db '1'

color db 1

data ends

code segment

（2）main函数，调用其他函数完成功能

main proc farassume ss:stack,ds:data,cs:code

mov ax,data

mov ds,ax

mov ax,stack

mov ss,ax

mov sp,32

lea dx,message1

call print\_str

lea dx,maxlen1

call input

lea dx,message2

call print\_str

lea dx,maxlen2

call input

call find\_str

lea dx,message3

call print\_str

call turn\_decimal

lea dx,message4

call print\_str

call show\_result

mov dl,0ah

mov ah,02h

int 21h

mov ah,4ch

int 21h

main endp

（3）print\_str函数，通过21中断输出到屏幕，调用前需将字符串首偏移地址存入dx(数据段message1~4)

print\_str proc near

mov ah,9

int 21h

ret

print\_str endp

（4）input函数，接收用户输入（母串，子串）函数，调用前需将接收输入区域的首偏移地址存入dx

input proc near

mov ah,0ah

int 21h

mov dl,13

mov ah,2

int 21h

mov dl,10

mov ah,2

int 21h

ret

input endp

（5）find\_str函数，依靠两层循环，外层循环枚举查找起点，内层循环比较字符，母串从该起点起，子串从头起，依次向后逐一对比。如果出现不匹配则内层循环退出， 外层循环进入下一轮；如果匹配了完整的子串，则说明在母串中成功找到了一个子串，在position中记录此次查找的起点，子串数目+1。下一次的查找起点需跳过子串长度个字符。直至从所有查找起点均查找过，外层循环退出。

find\_str proc near

lea si,string1

lea dx,position

mov al,len1 ;确定循环次数

sub al,len2

inc al

cbw

mov cx,ax

loop0:

lea di,string2

push cx

push si

mov al,len2

cbw

mov cx,ax

loop1:

mov bh,[si]

mov bl,[di]

cmp bh,bl

jne next0

inc si

inc di

loop loop1

inc decimal ;找到一个子串

pop si

mov di,dx

mov [di],si ;记录一个子串起始位置

push si

add dx,2

next0:

pop si

pop cx

inc si

loop loop0

ret

find\_str endp

（6）turn\_decimal函数，将需要转换的二进制数作为被除数（ax），要转换的进制（10）作为除数（dl），然后不断地进行除法，每做一次除法后，需将余数压入栈记录，然后用商作为下一轮除法的被除数（cbw）直至一次除法后商为0，结束除法过程。现在，栈中的每一个值代表着转换出的十进制数的一位（0~9），只要依次弹出栈顶值即可正确显示该十进制数。当然弹出过程中还需要将0~9数值转换为0~9字符。

turn\_decimal proc near

mov ax,decimal

mov dl,10

mov cx,0fffh

mov bx,sp ;记录栈顶位置，下面要向栈中压二进制转十进制的每一位，退栈时的边界

loop0: ;循环一次压入一位

div dl

add ah,30h

push ax ;余数在ah中，压入栈的每一字只有高字节为所需信息

cmp al,0

je next0

cbw

loop loop0

next0:

mov cx,0fffh

loop1: ;退栈，按位输出

pop dx

mov dl,dh ;将有效信息放入dl中等待输出

mov ah,02h

int 21h

cmp sp,bx

je next1

loop loop1

next1:

mov dl,13

mov ah,2

int 21h

mov dl,10

mov ah,2

int 21h

ret

turn\_decimal endp

（7）change\_color函数，共10种颜色，每连续10个子串使用10个不同颜色，之后颜色循环重复。每用颜色标注一个子串，就令color增1，然后将color对10取模的值映射到10个不同的颜色。

change\_color proc near

mov al,color

cbw

mov bl,10

div bl

inc ah

mov color,ah

ret

change\_color endp

（8）print\_color\_char函数，用来存放需要变色的字符，调用BIOS 09h号中断以特定样式显示color\_char中字符，之后依次调用BIOS 03h,02h中断将光标向后移动一格。

print\_color\_char proc near ;设置字符颜色闪烁

mov ah,9

mov al,color\_char ;al:要改变颜色的字符

mov cx,1 ;cx:字符重复次数

mov bl,color ;bl:颜色属性

add bl,80h ;+80变为闪烁

mov bh,0

int 10h

mov ah,3 ;读光标位置

mov bh,0

int 10h

inc dl ;光标位置后移

mov ah,2 ;置光标位置

int 10h

ret

print\_color\_char endp

（9）show\_result函数,显示结果。同时遍历母串（string1）和position数组，每遍历到一个子串的起始，开始用颜色标注接下来的子串长度（actlen2）个字符。将每一个子串字符存入color\_char然后调用print\_color\_char，每显示完一个子串调用change\_color以刷新颜色。

show\_result proc near

lea si,string1

lea di,position

mov al,len1

cbw

mov cx,ax

loop0:

cmp si,[di]

jne next0

push cx

mov al,len2 ;找到一个子串起始位置，从这里开始，子串长度个字符应用颜色闪烁标记

cbw

mov cx,ax

loop1:

mov al,[si]

mov color\_char,al

push cx

call print\_color\_char

pop cx

inc si

loop loop1

call change\_color ;刷新颜色

pop cx

sub cl,len2 ;跳过子串长度，继续向后查找子串起始位置

inc cx

add di,2

jmp next1

next0:

mov dl,[si]

mov ah,2

int 21h

inc si

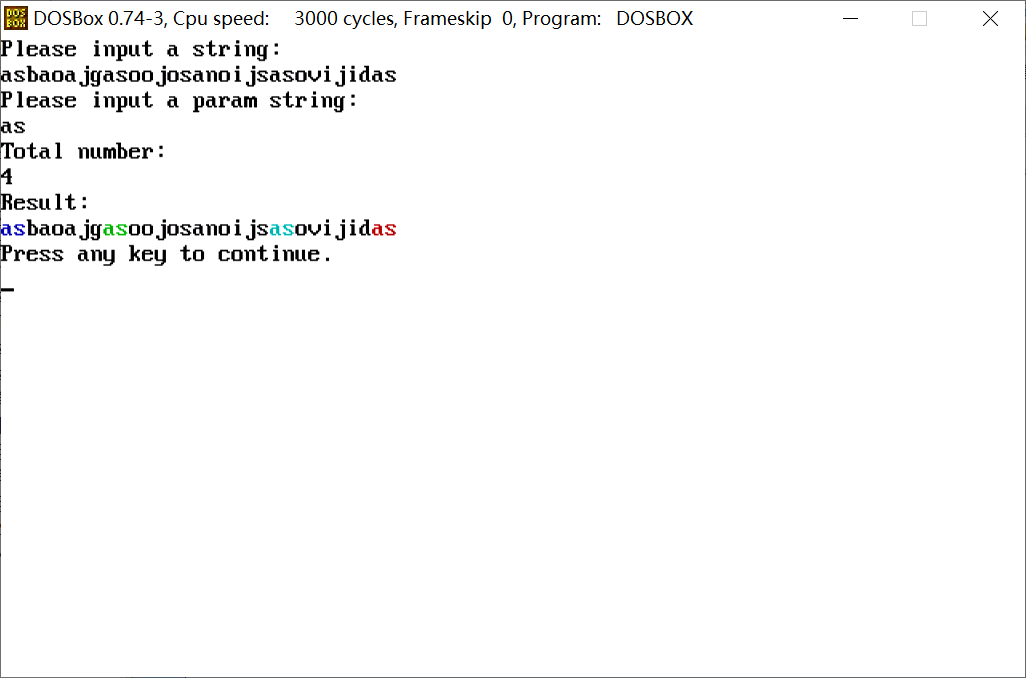
next1:

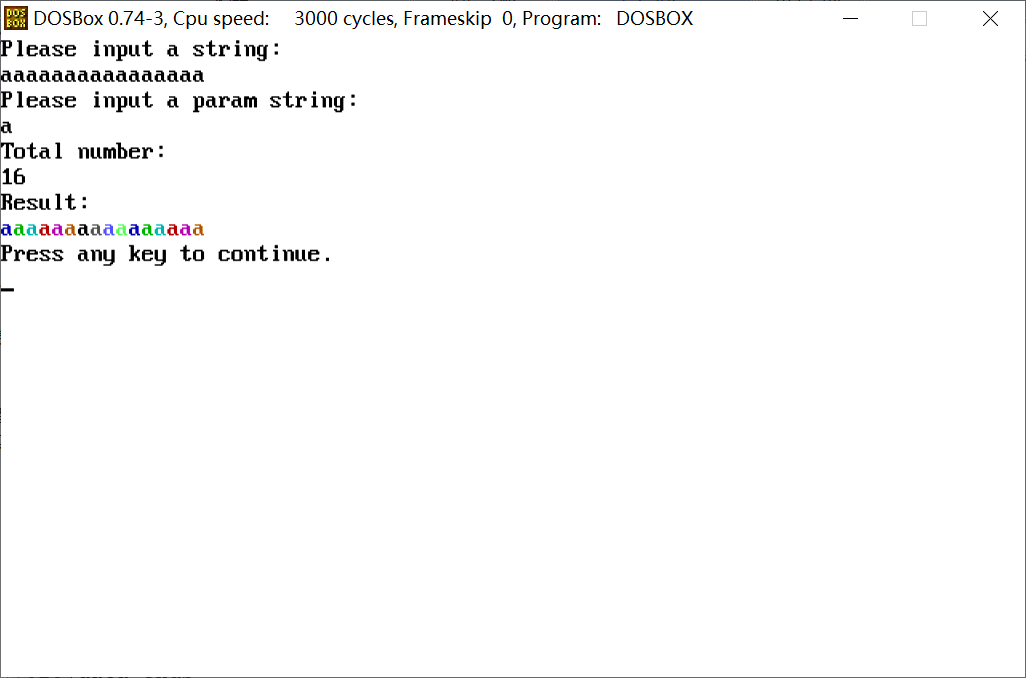
loop loop0

ret

show\_result endp

# 运行结果测试与分析：





# 5、结论与心得：

本次课设印象最深的就是在给老师检查时，老师跟我的谈话，问我的为什么输入只能输入那么多字符，我只能回答因为我试过所以知道，然后老师将其中字节和数据存储大小的关系给我讲透彻了，老师说我们不仅要知道他是这样，我们还要知道他为什么是这样，要知其然，并知其所以然。确实，在之前的实验中，我只是去简单了解了程序的大概作用，而没有认真思考这个程序为什么要这么做，只在一个特定的框架里搭建程序，没有想过这个框架是为什么搭建起来的。我希望在以后的工作中，要多思考更底层的原因，去更深刻的理解程序。