; 从键盘输入4\*4的矩阵

; （1）每个元素都是4位十进制数。

; （2）在屏幕上输出该矩阵和它的转置矩阵。

; （3）输出这两个矩阵的和（对应元素相加）。

; （4）求出该矩阵的鞍点（在行上大，在列上小）并在原矩阵中闪烁显示。

; （5）数据的输入和结果的输出都要有必要的提示，且提示独占一行。

; （6）要使用到子程序。

data segment ;数据段

str dw 10 dup(?)

str1 db 'Please input your matrix:',0ah,0dh,'$'

str2 db 'Your matrix is:',0ah,0dh,'$'

str3 db 'Your transpose matrix is:',0ah,0dh,'$'

str4 db 'The sum of matrices is:',0ah,0dh,'$'

crlf db 0ah,0dh,'$'

m1 dw 16 dup(?) ;原始矩阵

m2 dw 16 dup(?) ;转置矩阵

m3 dw 16 dup(?) ;矩阵之和

x dw 4 dup(?) ;记录行最大的下标

y dw 4 dup(?) ;记录列最小的下标

pos dw ? ;记录鞍点下标

data ends

stack segment stack ;堆栈段

dw 20 dup(?)

stack ends

code segment ;代码段

assume cs:code,ds:data,ss:stack

main proc far

start:

mov ax,data

mov ds,ax ;绑定ds到data

mov ah,09h

lea dx,str1

int 21h ;输出"Please input your matrix:"

call input\_data ;接收矩阵

mov ah,09h

lea dx,crlf

int 21h ;换行

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*闪烁鞍点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov ah,09h

lea dx,str2

int 21h ;输出“你的矩阵是:”

call saddle\_point

call print

lea bx,m1

call output\_data ;输出原矩阵（闪烁鞍点）

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*转置矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

call exchange ;计算转置矩阵

mov ah,09h

lea dx,str3 ;输出“矩阵的转置是:”

int 21h

lea bx,m2

call output\_data ;输出转置矩阵

mov ah,09h

lea dx,crlf

int 21h

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*计算矩阵和\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov ah,09h

lea dx,str4 ;输出“矩阵的和是:”

int 21h

call sum\_data ;计算矩阵和

lea bx,m3

call output\_data ;输出矩阵和

mov ax,4c00h

int 21h

main endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*编辑子程序\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*输入矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OK

input\_data proc near

mov di,0 ;记录当前这个数已经接收了多少位

mov si,0 ;记录当前接收到16个数中的第几个数

mov bx,0

L1:

;bx存已经存的数，al存着新接收的数，交换ax(拓展)和bx，ax\*10+bx存进bx中

mov ax,0

mov cx,0

mov dx,0

mov ah,01h

int 21h ;接收一个字符存进al中

inc di ;计数器记已经接收了多少位

sub al,30h ;转化为数值，存入ax中

cbw ;al拓展成字

xchg ax,bx ;交换内容

mov cx,10

mul cx

add bx,ax

mov cx,4 ;判断是否是四位数，是则存入

cmp di,cx

je L2

jmp L1

L2:

mov m1[si],bx

mov bx,0

mov ax,0

inc si

inc si ;双字加2

mov ah,02h

mov dl,' '

int 21h ;每输入一个四位数输出一个空格作划分

mov di,0 ;位数计数器清零

;x和4，8，12，16比较，4，8，12换行，16结束

cmp si,8

je L3

cmp si,16

je L3

cmp si,24

je L3

cmp si,32

je L4

;否则重新循环L1

jmp L1

L3:

lea dx,crlf;换行

mov ah,09h

int 21h

jmp L1

L4:

lea dx,crlf;换行

mov ah,09h

int 21h

ret

input\_data endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*输出原矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OK

output\_data proc near

mov si,0;当前输出第几位

O3:

mov cx,0;记录这个数有多少位

mov ax,[bx+si];要输出的数存ax作被除数

O1:

mov dx,0

mov di,10;除数为10

div di

push dx;余数压入栈中

inc cx

cmp ax,0

jnz O1;没除尽循环

O2:

pop dx;从高位开始输出

add dx,30h;数字转换成ascii码

mov ah,02h

int 21h

loop O2;cx控制循环

mov ah,02h

mov dl,' '

int 21h;输出空格隔开不同数字

add si,2

;输完一行换行

cmp si,8

je O4

cmp si,16

je O4

cmp si,24

je O4

;全部输完结束

cmp si,32

je O5

jmp O3

O4:

mov ah,09h

lea dx,crlf

int 21h

jmp O3

O5:

mov ah,09h

lea dx,crlf

int 21h

ret

output\_data endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*计算鞍点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OK

saddle\_point proc near

mov si,0;标记当前判断的位置

mov cx,0;标记行是否满四个

mov bp,0;标记第几个最值

;填写行最大下标数组x

mov di,0;bx存行最大值下标

row:

mov ax,m1[di]

cmp ax,m1[si]

jl change\_max

udr:

add si,2

inc cx

cmp cx,4

je row\_next;每处理四个数换下一行

jmp row

row\_next:

mov x[bp],di

mov di,si

mov cx,0

add bp,2

cmp bp,8

je row\_end;找到四个最大值结束

jmp row

change\_max:

mov di,si

jmp udr

row\_end:

mov si,0;标记当前判断的位置

mov cx,0;标记列是否满四个

mov bp,0;标记第几个最值

;填写列最小下标数组y

mov di,0;bx存列最小值下标

col:

mov ax,m1[si]

cmp ax,m1[di]

jl change\_min

udc:

mov ax,si

add ax,8

mov si,ax

inc cx

cmp cx,4

je col\_next;下一列

jmp col

col\_next:

mov cx,0

mov y[bp],di

add bp,2

cmp bp,8

je col\_end

mov si,bp

mov di,si

jmp col

change\_min:

mov di,si

jmp udc

col\_end:

;双重循环遍历x,y数组找鞍点

mov di,0;遍历x

mov si,0;遍历y

cir\_x:

cmp di,8

je over

mov si,0

cir\_y:

mov ax,x[di]

cmp ax,y[si]

je get

add si,2

cmp si,8

jl cir\_y

add di,2

jmp cir\_x

get:

mov ax,x[di]

mov pos,ax

over:

ret

saddle\_point endp

;输出鞍点

print proc near

sta:

mov cx,0;记录这个数有多少位

mov ax,pos;要输出的数存ax作被除数

fir:

mov dx,0

mov di,10;除数为10

div di

push dx;余数压入栈中

inc cx

cmp ax,0

jnz fir;没除尽循环

sec:

pop dx

add dx,30h

mov ah,02h

int 21h

loop sec

mov ah,09h

lea dx,crlf

int 21h

ret

print endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*转置矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

exchange proc near

mov si,0;标记m1遍历到的位置

mov di,0;标记m2遍历到的位置

mov cx,0;标记m1遍历到第几列

mov dx,0;行计数器，计算同一列的四行是不是都遍历完了

mov ax,0;作数据中转

E2:

mov ax,m1[si]

mov m2[di],ax

add si,8;+8移到同一列的下一个

inc di

inc di;+2移到同一行的下一个

inc dx;dx+1

cmp dx,4

je E1

jmp E2

E1:

mov dx,0;行计数器清零

inc cx

inc cx

mov si,cx

cmp cx,8

je E3

jmp E2

E3:

mov ah,09h

lea dx,crlf

int 21h

ret

exchange endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*计算矩阵和\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OK

sum\_data proc near;计算矩阵的和，结果转移到m3中

mov ax,0

mov si,0

S1:

mov ax,m1[si]

add ax,m2[si]

mov m3[si],ax

add si,2

cmp si,32

je S2

jmp S1

S2:

ret

sum\_data endp

code ends

end start