.model small

.stack 300h

.data

enter\_col db 'Enter number of columns: ', '$'

enter\_row db 'Enter number of rows: ', '$'

sums\_of\_col db 'Sums of columns: ', '$'

row db 0

col db 0

str db 100, ?, dup('$')

matrix db 100 dup(?)

buf db 4, 4, ' $'

enter db 0ah, 0dh, '$'

.code

start:

mov ax, @data

mov ds, ax

mov ah, 09h

mov dx, offset enter\_col

int 21h

mov ah, 01h

int 21h

sub al, 30h

mov col, al

mov ah, 09h

lea dx, enter

int 21h

mov dx, offset enter\_row

int 21h

mov ah, 01h

int 21h

sub al, 30h

mov row, al

mov ah, 09h

lea dx, enter

int 21h

mov ah, 0Ah

lea dx, str

int 21h

mov ah, 09h

lea dx, enter

int 21h

int 21h

xor ax,ax

xor cx,cx

mov di, offset [str + 2]

mov cl, [str + 1]

mov bl, 10

xor si, si

str\_to\_num:

mov dl, [di]

cmp dl, ' '

je space

sub dl, 30h

mul bl

add al, dl

inc di

loop str\_to\_num

cmp dl, ' '

jne add\_to\_array

jmp next\_step

space:

mov matrix[si], al

xor ax, ax

inc di

inc si

loop str\_to\_num

jmp next\_step

add\_to\_array:

mov matrix[si], al

xor ax, ax

inc si

next\_step:

mov cl, col

find\_sum:

xor ax, ax

push cx

mov bh, cl

mov cl, row

dec bh

xor dx, dx

count\_col:

mov al, cl

dec al

mul col

add al, bh

mov di, ax

add dl, matrix[di]

loop count\_col

pop cx

push dx

loop find\_sum

mov cx, si

xor si, si

mov bh, 0

num\_to\_str:

add bh, 1

xor ax, ax

mov al, matrix[si]

mov di, offset buf[4]

get\_num:

xor ah, ah

div bl

add ah, 30h

mov [di], ah

dec di

cmp al,0

jne get\_num

inc di

mov dx, offset [buf + 2]

mov ah, 09h

int 21h

cmp bh, col

jne space\_

mov bh, 0

mov ah, 09h

lea dx, enter

int 21h

jmp buf\_delete

space\_:

mov ah, 02h

mov dx, ' '

int 21h

buf\_delete:

mov [di], ' '

inc di

cmp [di],'$'

jne buf\_delete

inc si

loop num\_to\_str

mov cl, col

mov ah, 09h

lea dx, enter

int 21h

lea dx, sums\_of\_col

int 21h

get\_sum:

xor di, di

pop ax

sum\_to\_str:

xor dx,dx

div bx

add dx, 30h

push dx

inc di

cmp ax,0

jne sum\_to\_str

mov si, cx

mov cx, di

mov ah, 02h

print:

pop dx

int 21h

loop print

mov dx, ' '

int 21h

mov cx, si

loop get\_sum

mov ah, 00ch

int 21h

end start

.model small

.stack 100h

.data

string db 30, ?, dup('$')

.code

start:

mov ax, @data

mov ds, ax

mov ah, 0ah

mov dx, offset string

int 21h

xor ax, ax

xor cx, cx

mov si, offset string[string+2]

mov cl, string+1

mov di, si

mov al, string[di]

change\_symb:

cmp cx, 0

je print

inc si

cmp string[si], 0Dh

je swap

cmp string[si], ' '

je swap

loop change\_symb

swap:

mov bl, string[si-1]

mov string[di], bl

mov string[si-1], al

cmp string[si], 0Dh

je print

inc si

mov di, si

cmp string[si], ' '

je change\_symb

mov al, string[si]; если не пробел после пробела то начинается следуюшеее слово

loop change\_symb

print:

mov cl, string+1

mov si, offset string[string+2]

xor ax,ax

xor bx, bx

print\_symb:

mov dl, string[si]

mov ah, 02h

int 21h

inc si

loop print\_symb

mov ah, 00ch

int 21h

end start

.model small

.stack 100h

.data

str db 30, ?, dup ('$')

array db 10 dup (?)

.code

start:

mov ax,@data

mov ds, ax

mov ah, 0ah

mov dx, offset str

int 21h

xor ax,ax

xor cx,cx

mov di, offset [str+2]

mov cl, str+1

mov bl, 10

xor si, si

str\_to\_num:

mov dl, [di]

cmp dl, ' '

je space

sub dl, 30h

mul bl

add al, dl

inc di

loop str\_to\_num

cmp dl, ' '

jne add\_to\_array

jmp next\_step ;переход

space:

mov array[si], al

xor ax, ax

inc di

inc si

loop str\_to\_num

add\_to\_array: ; из-за того что space не добавляется послед число вручную

mov array[si], al ; без пробела в конце запись массива

xor ax, ax

inc si

xor dx,dx

mov cx, si

sort:

cmp cx, 1

je next\_step

push si

sub si, cx

push cx

mov dl, array[si]

mov di, si ; первый элем как мин

inc si

dec cx

find\_min:

cmp dl, array[si]

jg save\_min

inc si

loop find\_min

jmp swap

save\_min:

mov dl, array[si] ; dl - новый мин элем

mov di, si

inc si

loop find\_min

swap:

pop cx

pop si

push si

sub si, cx

mov bl, array[si]

mov array[si], dl

mov array[di], bl

pop si

loop sort

next\_step:

mov cx, si

xor si, si

extract\_num\_array:

mov al, array[si]

xor di, di ; сколько цифр в стеке

xor ah,ah

mov bx, 10

num\_to\_str:

div bl

add ah, 30h

mov dl, ah

push dx ; в обратном порядке ввыодит

xor ah,ah

inc di

cmp al, 0

jne num\_to\_str

print:

pop dx

mov ah, 02h

int 21h

dec di

cmp di,0

jne print

inc si ; переход к следующему числу в масс

mov ah, 02h

mov dx, ' '

int 21h

loop extract\_num\_array

mov ah, 00ch

int 21h

end start

.model small

.stack 100h

.data

sum dw 0

str db 30, ?, dup('$')

.code

start:

mov ax,@data

mov ds, ax

mov ah, 0ah

mov dx, offset str

int 21h

xor ax,ax

xor cx,cx

lea di, [str+2]

mov cl, str+1

mov bx, 10

str\_to\_num:

mul bx

mov dl, [di]

sub dl, 30h

add ax, dx

inc di

push dx ; для суммы

loop str\_to\_num

mov cl, str+1

xor dx, dx

sum\_of\_num:

pop bx

add dx, bx ; всю сумму сразу чтобы записать

loop sum\_of\_num

mov sum, dx

mov cl, str+1

xor dx, dx

mov bx, 10

num\_to\_str:

div bx

add dl, 30h

push dx

xor dx, dx

inc di

loop num\_to\_str

mov cl, str+1

xor dx, dx

print:

pop dx

mov ah, 02h

int 21h

loop print

xor dx, dx

mov ax, sum

num\_to\_str\_sum:

div bx

add dl, 30h

push dx

xor dx, dx

add cx, 1 ; для print sum

cmp ax, 0

je print\_sum

jmp num\_to\_str\_sum ; бесконечный цикл из-за того что не знаю кол-во разрядов суммы

print\_sum:

pop dx

mov ah, 02h

int 21h

loop print\_sum

mov ah, 00ch

int 21h

end start