;program to implement bubble sort algorithm

%macro write 2

PUSHA

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

POPA

%endmacro

%macro read 2

PUSHA

mov eax,3

mov ebx,0

mov ecx,%1

mov edx,%2

int 80h

POPA

%endmacro

section .data

msg db "Enter the number of elements : "

msglen equ $-msg

msg2 db "Enter the Elements : ", 13, 10

msglen2 equ $-msg2

msg3 db "Array Elements : ", 13, 10

msglen3 equ $-msg3

newline db 10, 13

space db " "

section .bss

arr resb 10

size resb 2

num1 resb 2

temp resb 2

section .text

global \_start

\_start:

write msg, msglen

read num1, 2

read size, 1

xor eax,eax

xor ebx,ebx

xor ecx,ecx

CALL convert

MOV [size], ebx

MOV ecx, [size]

MOV edi, arr

write msg2, msglen2

again:

read edi, 1

read edx, 1

sub byte[edi], '0'

inc eax

inc edi

dec ecx

jnz again

CALL bubble

mov eax, 0

mov ecx, [size]

MOV edi, arr

write msg3, msglen3

again1:

add byte[edi], '0'

write edi, 1

write space, 1

inc eax

inc edi

dec ecx

jnz again1

MOV eax, 1

int 80h

convert :

mov esi ,num1

mov edi ,size

mov cl ,02h

xor eax,eax

xor ebx,ebx

up : rol bl,04h

mov al,[esi]

cmp al,39h

jbe skipc

sub al,07h

skipc :

sub al,30h

add bl,al

mov [edi],bl

inc esi

inc edi

loop up

ret

bubble:

mov ch, [size]

mov esi, arr

;INC ESI

mov ah, [size]

loop1:

mov cl, [size]

mov edi, esi

dec cl

loop2:

mov bh, [edi]

mov dh, [edi + 1]

CMP bh, dh

JBE noswap

mov [edi], dh

mov [edi + 1], bh

noswap:

inc ah

inc edi

dec cl

jnz loop2

dec ch

jnz loop1

ret

OUTPUT:

