**Code:**

%macro write 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 0x80

%endmacro

%macro read 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 0x80

%endmacro

section .data

input1 db "Enter number of elements: "

ilen1 equ $-input1

input2 db "Enter elements of array: ",10

ilen2 equ $-input2

input3 db "Enter element to search: ",10

ilen3 equ $-input3

output1 db "Array elements are: ",10

olen1 equ $-output1

output2 db "Element found at index:"

olen2 equ $-output2

output3 db "Element not found.",10

olen3 equ $-output3

array times 20 db 0

newline db 10

spa db " "

spalen equ $-spa

section .bss

num resb 4

i resb 4

elem resb 4

temp resb 4

pos resb 4

section .text

global \_start

\_start:

write input1,ilen1

read num,4

write input2,ilen2

mov byte[i],0

mov esi,array

input:

read temp,2

mov ebx,[temp]

mov [esi],ebx

inc esi

inc byte[i]

mov al,[i]

mov bl,[num]

sub bl,'0'

cmp al,bl

je exit\_input

jmp input

exit\_input:

write input3,ilen3

read elem,4

write output1,olen1

mov byte[i],0

mov esi,array

display:

mov ebx,[esi]

mov [temp],ebx

write temp,1

write spa,spalen

inc esi

inc byte[i]

mov al,[i]

mov bl,[num]

sub bl,'0'

cmp al,bl

jl display

write newline,1

mov byte[i],0

mov esi,array

check:

mov ebx,[esi]

mov [temp],ebx

mov bl,[temp]

mov al,[elem]

cmp al,bl

je found

jmp not\_found

found:

inc byte[i]

mov al,[i]

add al,'0'

mov [pos],al

dec byte[pos]

write output2,olen2

write pos,1

write newline,1

jmp exit

not\_found:

inc esi

inc byte[i]

mov al,[i]

mov bl,[num]

sub bl,'0'

cmp al,bl

jl check

je failed

failed:

write output3,olen3

write newline,1

jmp exit

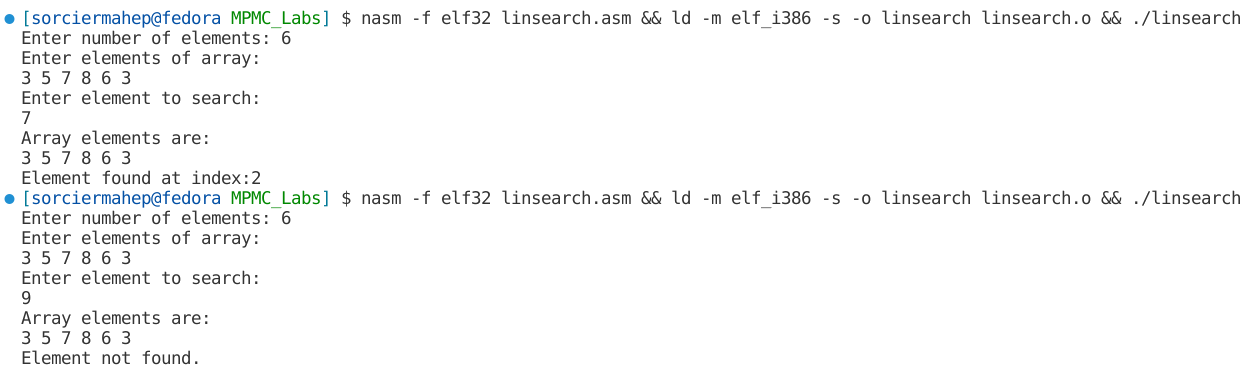
exit:

mov eax,1

xor ebx,ebx

int 0x80

**Output:**

****

**Code:**

%macro write 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 0x80

%endmacro

%macro read 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 0x80

%endmacro

section .data

input1 db "Enter number of elements: "

ilen1 equ $-input1

input2 db "Enter elements of array: ",10

ilen2 equ $-input2

input3 db "Enter element to search: ",10

ilen3 equ $-input3

output1 db "Array elements are: ",10

olen1 equ $-output1

output2 db "Element found at index:"

olen2 equ $-output2

output3 db "Element not found.",10

olen3 equ $-output3

array times 10 db 0

newline db 10

spa db " "

spalen equ $-spa

section .bss

num resb 4

elem resb 4

index resb 4

lb resb 1

ub resb 1

mid resb 1

trash resb 1

section .text

global \_start

\_start:

write input1,ilen1

read num,1

read trash,1

sub [num],byte '0'

write input2,ilen2

mov eax,array

mov edx,[num]

call input

write output1,olen1

mov eax,array

mov edx,[num]

call display

write input3,ilen3

read elem,1

read trash,1

mov eax,array

mov edx,[num]

mov edi,[elem]

call binsearch

mov eax,1

xor ebx,ebx

int 0x80

;eax contains address of array

;edx contains size of array

;edi contains number to search

input:

mov ecx,0

repeat\_input:

cmp ecx,edx

jge after\_input

mov esi,eax

add esi,ecx

pushad

read esi,1

read trash,1

popad

inc ecx

jmp repeat\_input

after\_input:

ret

display:

mov ecx,0

repeat\_display:

cmp ecx,edx

jge after\_display

mov esi,eax

add esi,ecx

pushad

write esi,1

write spa,spalen

popad

inc ecx

jmp repeat\_display

after\_display:

write newline,1

ret

binsearch:

and edi,000fh

mov [lb],byte 0

mov [ub],dl

repeat\_search:

pushad

mov al,[lb]

add al,[ub]

cbw ;convert byte to word

mov bl,2

div bl

mov [mid],al

popad

mov cl,[lb]

mov dl,[ub]

cmp cl,dl

jg not\_found

mov edx,[mid]

and edx,000fh

mov esi,dword [eax+edx]

and esi,000fh

cmp edi,esi

je found

jl lower\_part

upper\_part:

mov bl,[mid]

add bl,1

mov [lb],bl

jmp repeat\_search

lower\_part:

mov bl,[mid]

sub bl,1

mov [ub],bl

jmp repeat\_search

found:

add edx,'0'

mov [index],edx

pushad

write output2,olen2

write index,1

write newline,1

popad

ret

not\_found:

write output3,olen3

ret

**Output:**

