9A] Write an assembly language program to implement Linear Search.

**Program:**

section .data

prompt db "Enter number of elements",10

plen equ $-prompt

prompt2 db "Enter elements",10

p2len equ $-prompt2

prompt3 db "Elements are:",10

p3len equ $-prompt3

prompt4 db "Enter element to search:",10

p4len equ $-prompt4

prompt5 db "Found at index: "

p5len equ $-prompt5

prompt6 db "Not found: "

p6len equ $-prompt6

newline db 10

space db ' '

section .bss

n resb 4

num resb 4

arr resb 10

index resb 4

trash resb 1

%macro SYS\_WRITE 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

%macro SYS\_READ 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

mov eax,3

mov ebx,2

mov ecx,trash

mov edx,1

int 80h

%endmacro

section .text

global \_start

\_start:

SYS\_WRITE prompt,plen

SYS\_READ n,1

sub [n],byte '0'

SYS\_WRITE prompt2,p2len

mov eax,arr

mov edx,[n]

call input

SYS\_WRITE prompt3,p3len

mov eax,arr

mov edx,[n]

call display

SYS\_WRITE prompt4,p4len

SYS\_READ num,1

mov eax,arr

mov edx,[n]

mov edi,[num]

call linear\_search

mov eax,1

mov ebx,0

int 80h

input:

;; eax should contain address of array

;; edx should contain size of array

mov ecx,0

repeat\_input:

cmp ecx,edx

jge after\_input

mov esi,eax

add esi,ecx

pushad

SYS\_READ esi,1

popad

inc ecx

jmp repeat\_input

after\_input:

ret

display:

;; eax should contain address of array

;; edx should contain size of array

mov ecx,0

repeat\_display:

cmp ecx,edx

jge after\_display

mov esi,eax

add esi,ecx

pushad

SYS\_WRITE esi,1

SYS\_WRITE space,1

popad

inc ecx

jmp repeat\_display

after\_display:

mov eax,4

mov ebx,1

mov ecx,newline

mov edx,1

int 80h

ret

linear\_search:

;; eax should contain address of array

;; edx should contain size of array

;; edi should contain number to search

and edi,000fh

mov ecx,0

repeat\_search:

cmp ecx,edx

jge after\_search

mov esi,eax

add esi,ecx

mov ebx,[esi]

and ebx,000fh

cmp ebx,edi

jne not\_match

;;matched

add ecx,'0'

mov [index],ecx

pushad

SYS\_WRITE prompt5,p5len

SYS\_WRITE index,1

popad

ret

not\_match:

inc ecx

jmp repeat\_search

after\_search:

SYS\_WRITE prompt6,p6len

ret

**Output:**



9B] Write an assembly language program to implement Binary Search.

**Program:**

section .data

prompt db "Enter number of elements",10

plen equ $-prompt

prompt2 db "Enter elements",10

p2len equ $-prompt2

prompt3 db "Elements are:",10

p3len equ $-prompt3

prompt4 db "Enter element to search:",10

p4len equ $-prompt4

prompt5 db "Found at index: "

p5len equ $-prompt5

prompt6 db "Not found",10

p6len equ $-prompt6

newline db 10

space db ' '

section .bss

n resb 4

num resb 4

arr resb 10

index resb 4

lb resb 1

ub resb 1

mid resb 1

trash resb 1

%macro SYS\_WRITE 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

%macro SYS\_READ 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

mov eax,3

mov ebx,2

mov ecx,trash

mov edx,1

int 80h

%endmacro

section .text

global \_start

\_start:

SYS\_WRITE prompt,plen

SYS\_READ n,1

sub [n],byte '0'

SYS\_WRITE prompt2,p2len

mov eax,arr

mov edx,[n]

call input

SYS\_WRITE prompt3,p3len

mov eax,arr

mov edx,[n]

call display

SYS\_WRITE prompt4,p4len

SYS\_READ num,1

mov eax,arr

mov edx,[n]

mov edi,[num]

call binary\_search

mov eax,1

mov ebx,0

int 80h

input:

;; eax should contain address of array

;; edx should contain size of array

mov ecx,0

repeat\_input:

cmp ecx,edx

jge after\_input

mov esi,eax

add esi,ecx

pushad

SYS\_READ esi,1

popad

inc ecx

jmp repeat\_input

after\_input:

ret

display:

;; eax should contain address of array

;; edx should contain size of array

mov ecx,0

repeat\_display:

cmp ecx,edx

jge after\_display

mov esi,eax

add esi,ecx

pushad

SYS\_WRITE esi,1

SYS\_WRITE space,1

popad

inc ecx

jmp repeat\_display

after\_display:

mov eax,4

mov ebx,1

mov ecx,newline

mov edx,1

int 80h

ret

binary\_search:

;; eax should contain address of array

;; edx should contain size of array

;; edi should contain number to search

and edi,000fh

mov [lb],byte 0 ;lb=0

mov [ub],dl ;ub=n

repeat\_search:

pushad

mov al,[lb] ;mid=(lb+ub)/2

add al,[ub]

cbw

mov bl,2

div bl

mov [mid],al

popad

;;----------------------------

mov cl,[lb]

mov dl,[ub] ;we don't need value of n anymore

cmp cl,dl

jg after\_search ;exit loop if lb>ub

;;----------------------------

mov edx,[mid]

and edx,000fh

mov esi,dword [eax+edx]

and esi,000fh ;esi=arr[mid]

cmp edi,esi

je matched ;num == arr[mid]

jl lower\_part ;num < arr[mid]

upper\_part:

mov bl,[mid]

add bl,1

mov [lb],bl ;lb=mid+1

jmp repeat\_search

lower\_part:

mov bl,[mid]

sub bl,1

mov [ub],bl ;ub=mid-1

jmp repeat\_search

matched:

;display result

add edx,'0'

mov [index],edx

pushad

SYS\_WRITE prompt5,p5len

SYS\_WRITE index,1

popad

ret

after\_search:

;display not found

SYS\_WRITE prompt6,p6len

ret

**Output:**

