VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessors and Interfacing

(CSE – 3002)

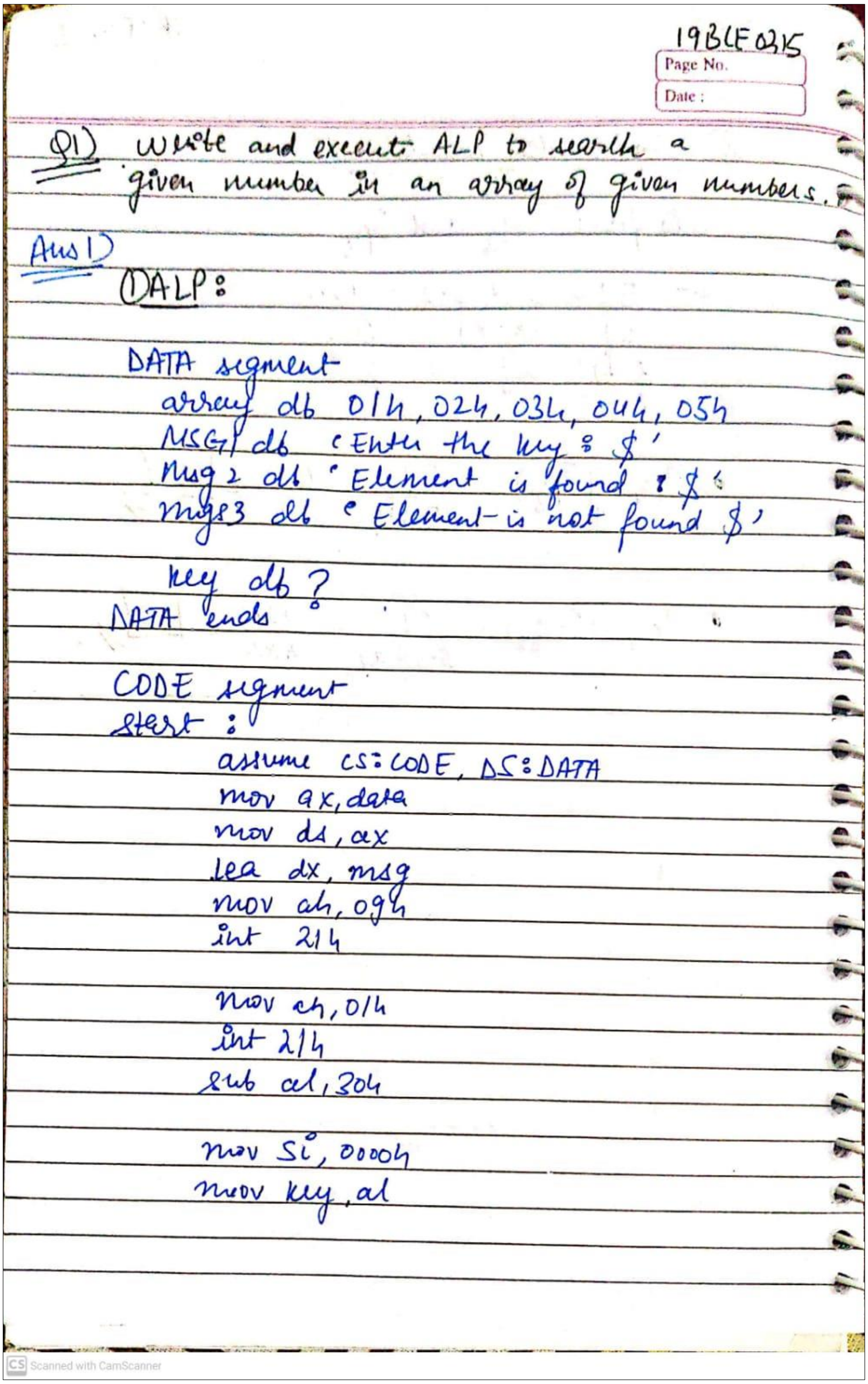
LAB EXPERIMENT– 3

Name: **Vibhu Kumar Singh**

Reg. No: **19BCE0215**

Teacher: **Mr. Konguvel E.**

1. Write and execute ALP to search a given number in an array of given numbers.



19BCE0315

Page No.

Date :

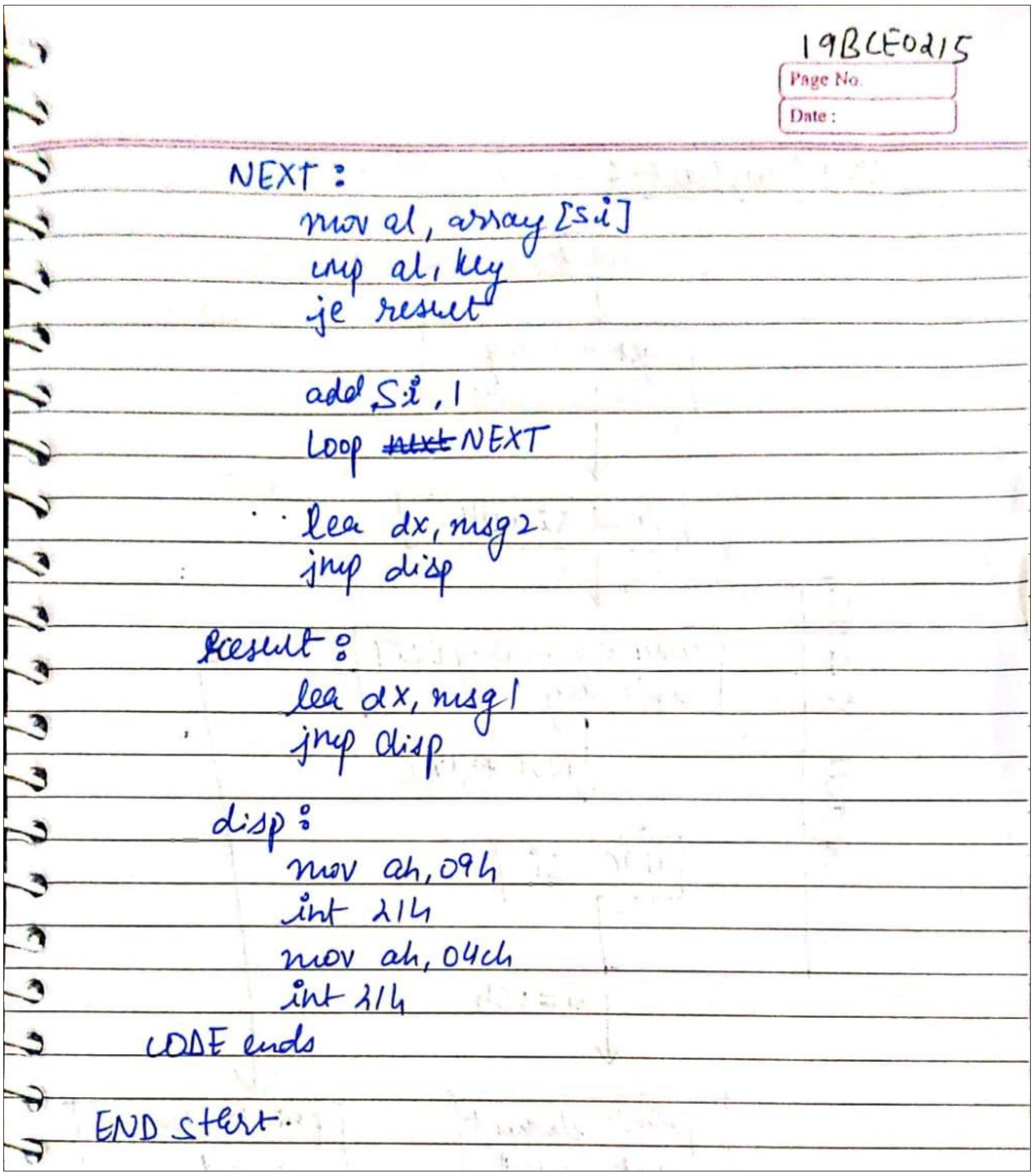
Q1) write and execute ALP to search a given number in an array of given numbers.

Ans1)

1) ALP:

```
DATA segment
  array db 01h,02h,03h,04h,05h
  MSG1 db 'Enter the key : $'
  Msg2 db 'Element is found : $'
  msg3 db 'Element is not found $'

  key db ?
DATA ends

CODE segment
start :
        assume CS:CODE, DS:DATA
        mov ax,data
        mov ds,ax
        lea dx, msg
        mov ah,09h
        int 21h

        mov ah,01h
        int 21h
        sub al,30h

        mov si,0000h
        mov key,al
```

Scanned with CamScanner



19BCE0215

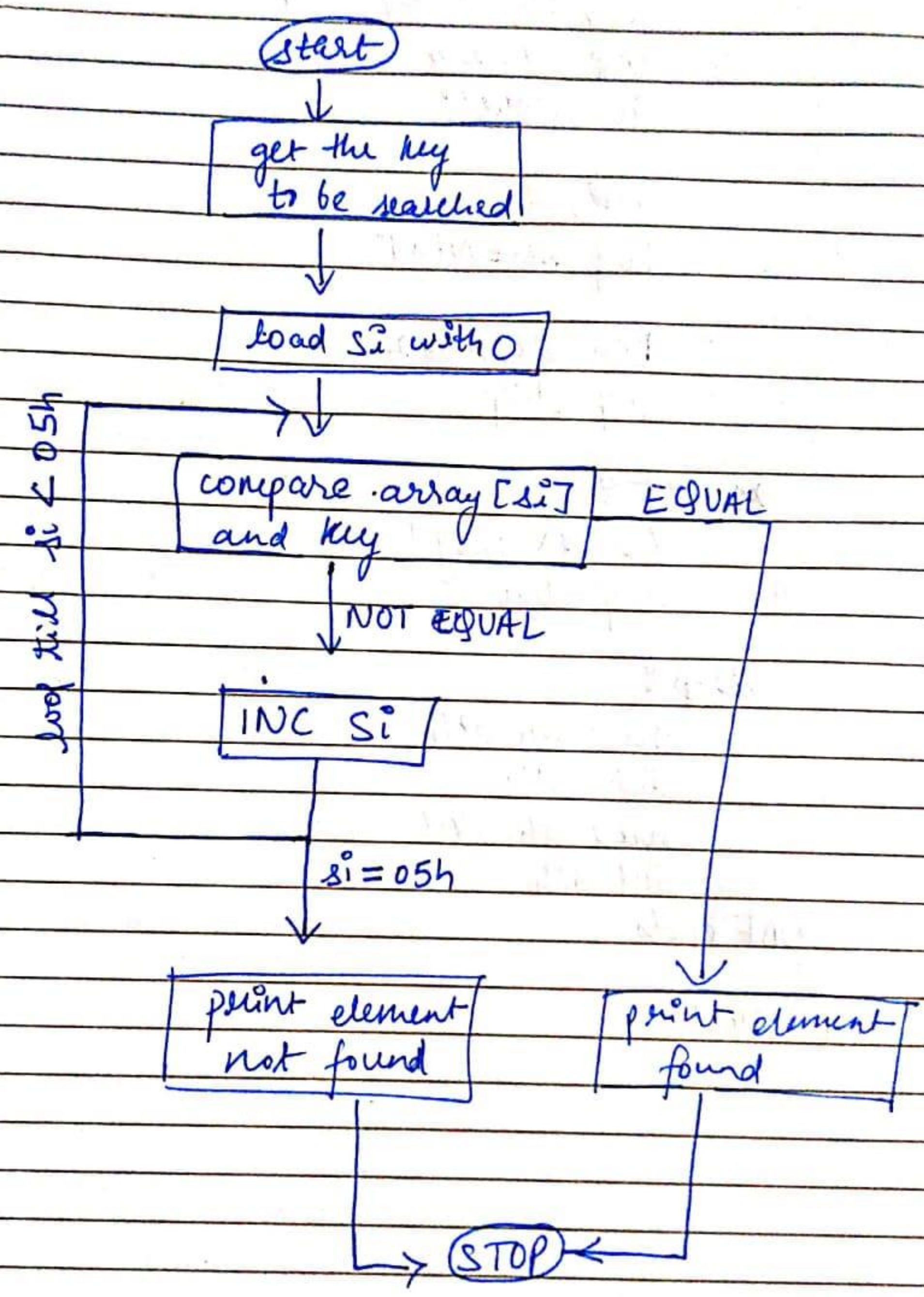
Page No.

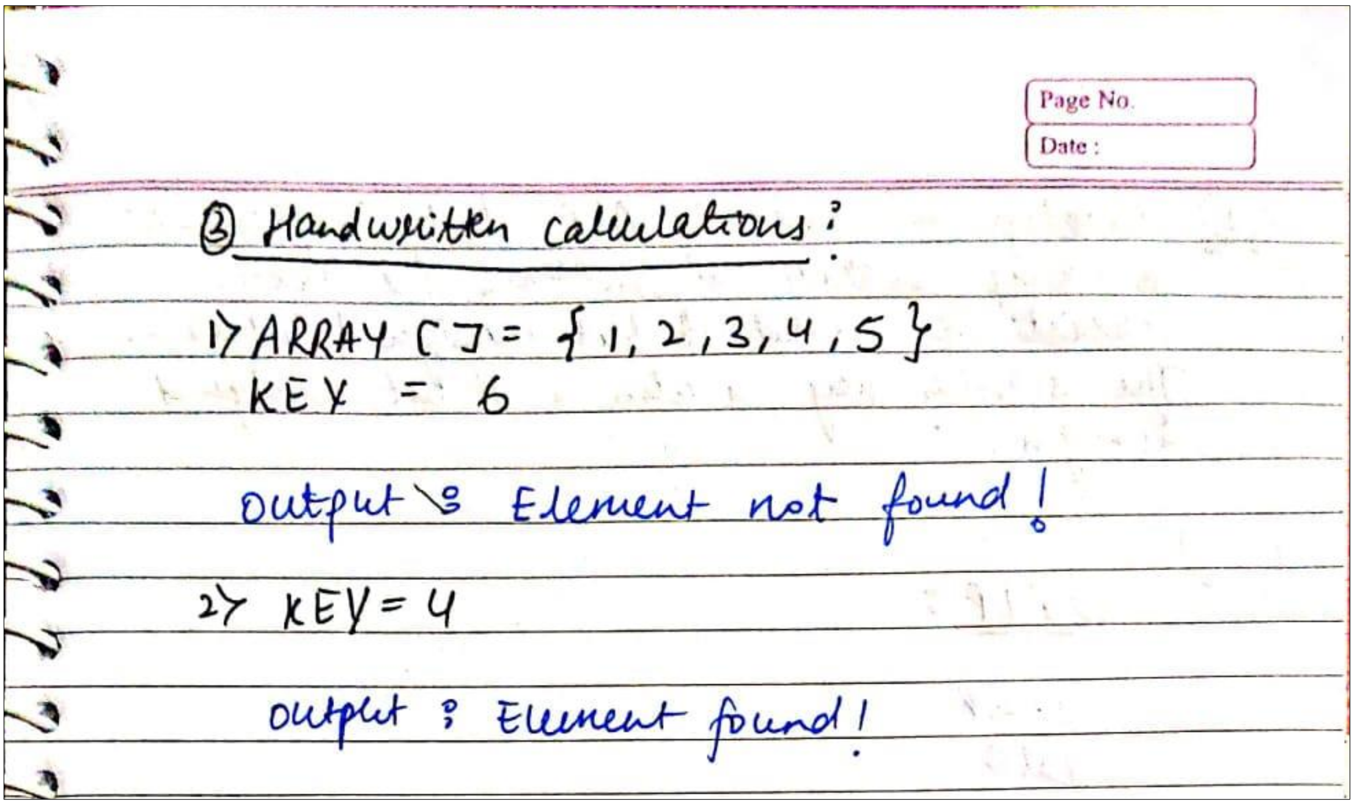
Date :

```
    NEXT :
        mov al, array[si]
        cmp al, key
        je reset

        add si, 1
        Loop ~~next~~ NEXT

        lea dx, msg2
        jmp disp

    Reset :
        lea dx, msg1
        jmp disp

    disp :
        mov ah,09h
        int 21h
        mov ah,04ch
        int 21h
  CODE ends

END start.
```

Page No.
Date :

② Flowchart :




Page No.

Date :

③ Handwritten calculations :

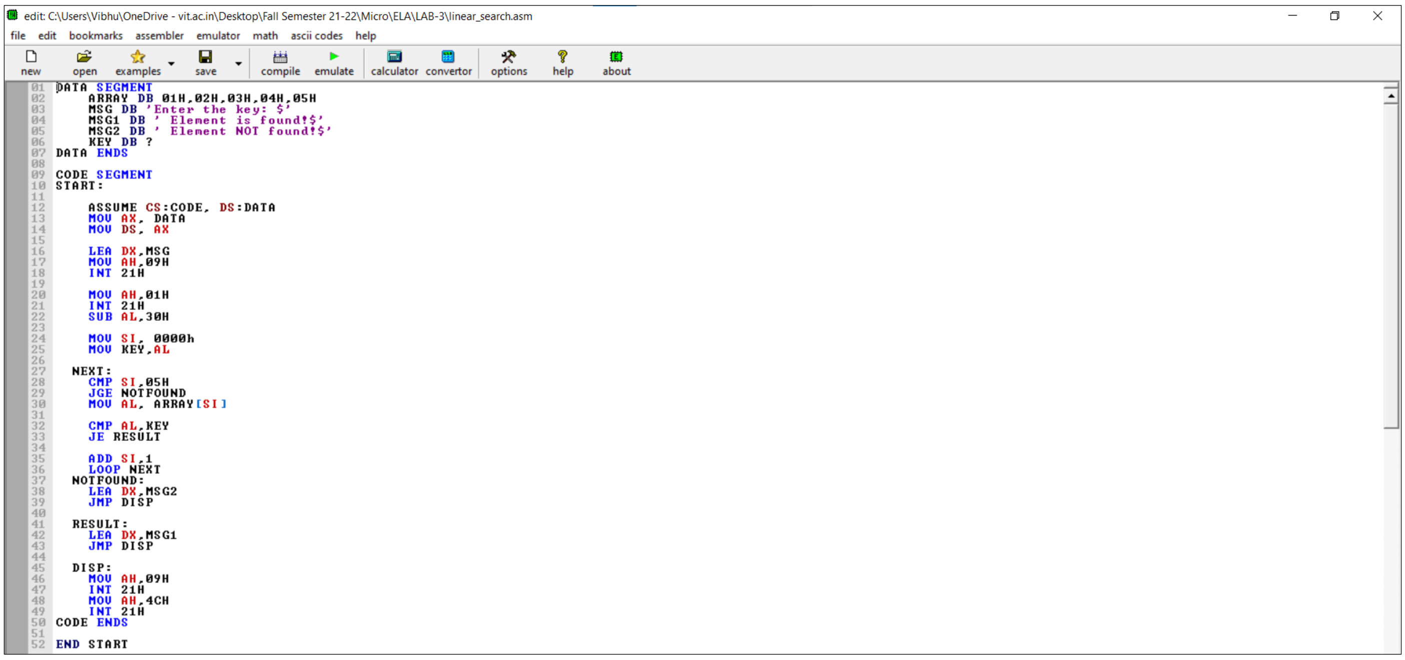
1) ARRAY [ ] = { 1, 2, 3, 4, 5 }
KEY = 6

output : Element not found!

2) KEY = 4

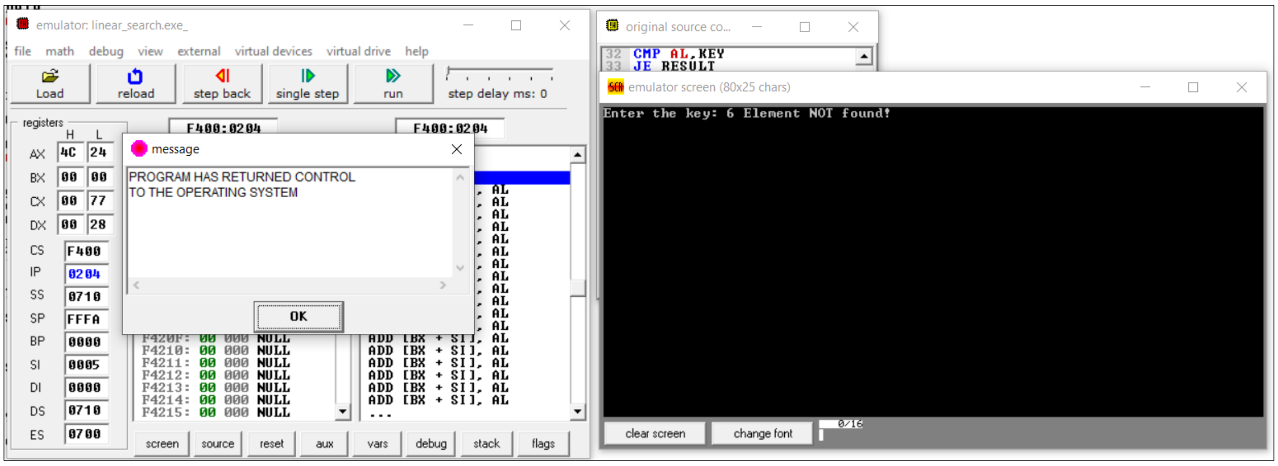
output : Element found!

Screenshot of ALP:


edit: C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\Fall Semester 21-22\Micro\ELA\LAB-3\linear_search.asm

```
DATA SEGMENT
    ARRAY DB 01H,02H,03H,04H,05H
    MSG DB 'Enter the key: $'
    MSG1 DB ' Element is found!$'
    MSG2 DB ' Element NOT found!$'
    KEY DB ?
DATA ENDS

CODE SEGMENT
START:

    ASSUME CS:CODE, DS:DATA
    MOV AX, DATA
    MOV DS, AX

    LEA DX,MSG
    MOV AH,09H
    INT 21H

    MOV AH,01H
    INT 21H
    SUB AL,30H

    MOV SI, 0000h
    MOV KEY,AL

  NEXT:
    CMP SI,05H
    JGE NOTFOUND
    MOV AL, ARRAY[SI]

    CMP AL,KEY
    JE RESULT

    ADD SI,1
    LOOP NEXT
  NOTFOUND:
    LEA DX,MSG2
    JMP DISP

  RESULT:
    LEA DX,MSG1
    JMP DISP

  DISP:
    MOV AH,09H
    INT 21H
    MOV AH,4CH
    INT 21H
CODE ENDS

END START
```

Screenshot of Output:

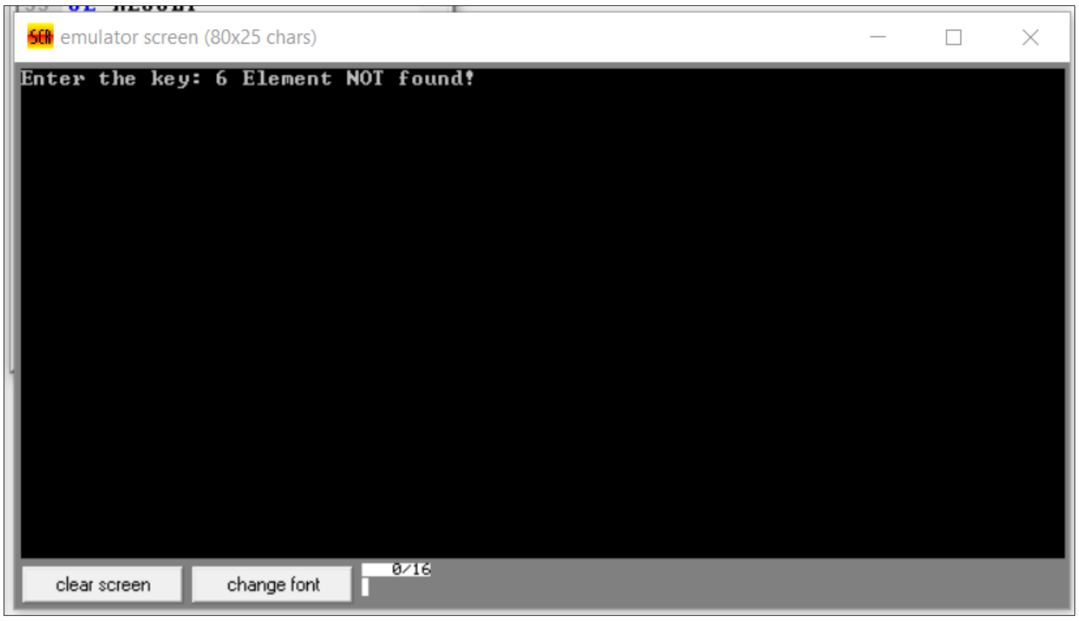


```
CMP AL,KEY
JE RESULT
```

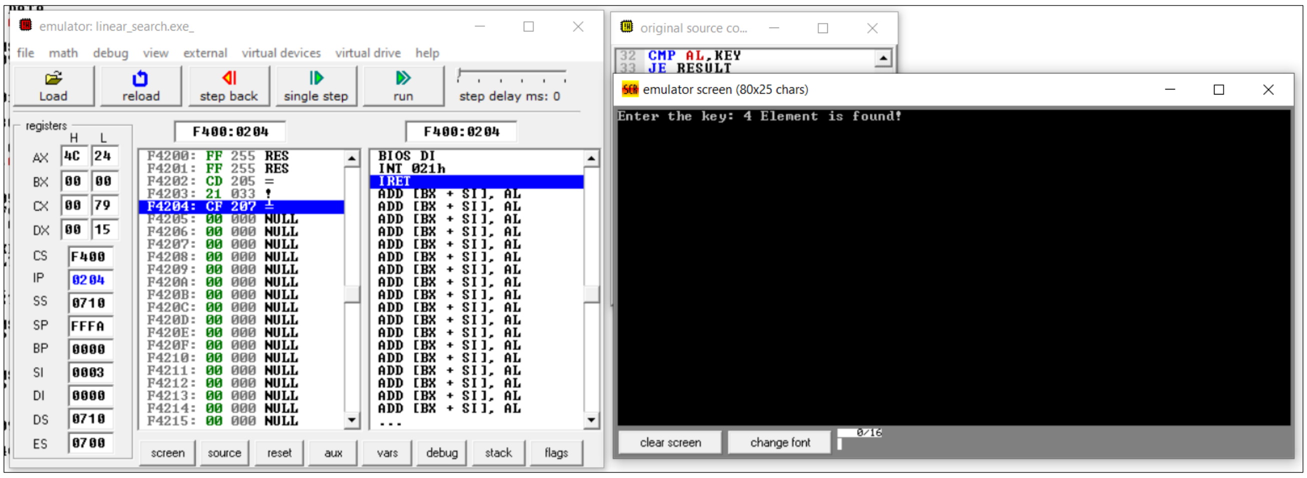
```
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
```

```
AX 4C 24   BX 00 00   CX 00 77   DX 00 28
CS F400  IP 0204  SS 0710  SP FFFA  BP 0000
SI 0005  DI 0000  DS 0710  ES 0700
```

```
Enter the key: 6 Element NOT found!
```



```
Enter the key: 6 Element NOT found!
```

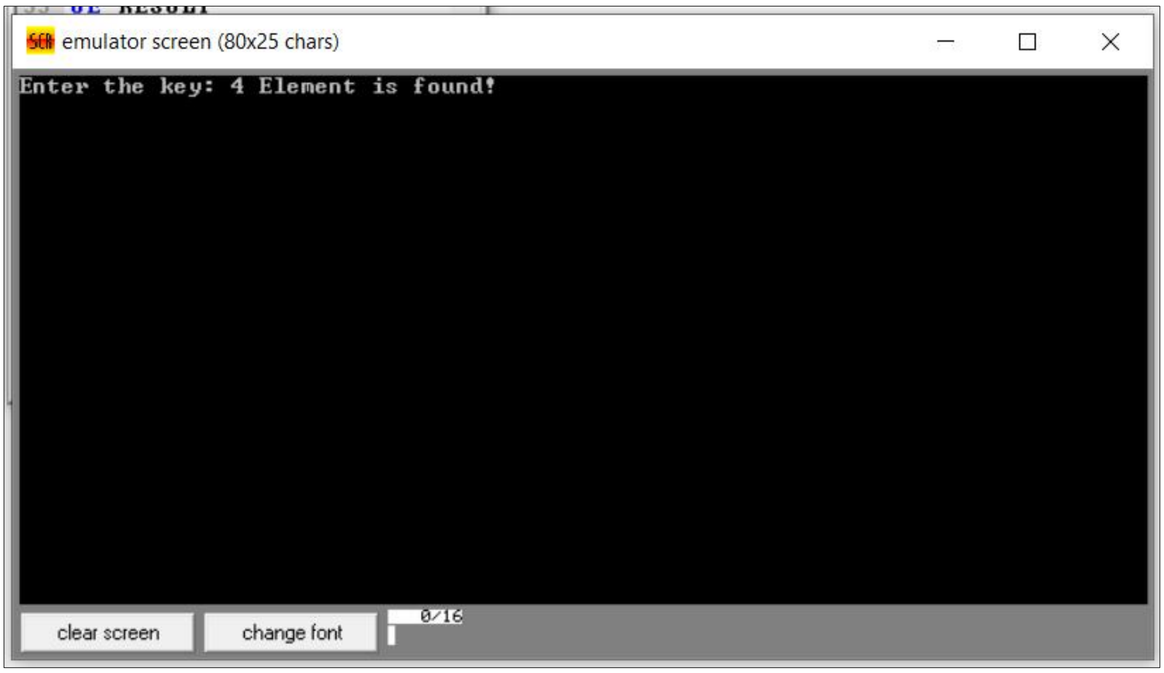


```
CMP AL,KEY
JE RESULT
```

```
AX 4C 24   BX 00 00   CX 00 79   DX 00 15
CS F400  IP 0204  SS 0710  SP FFFA  BP 0000
SI 0003  DI 0000  DS 0710  ES 0700
```

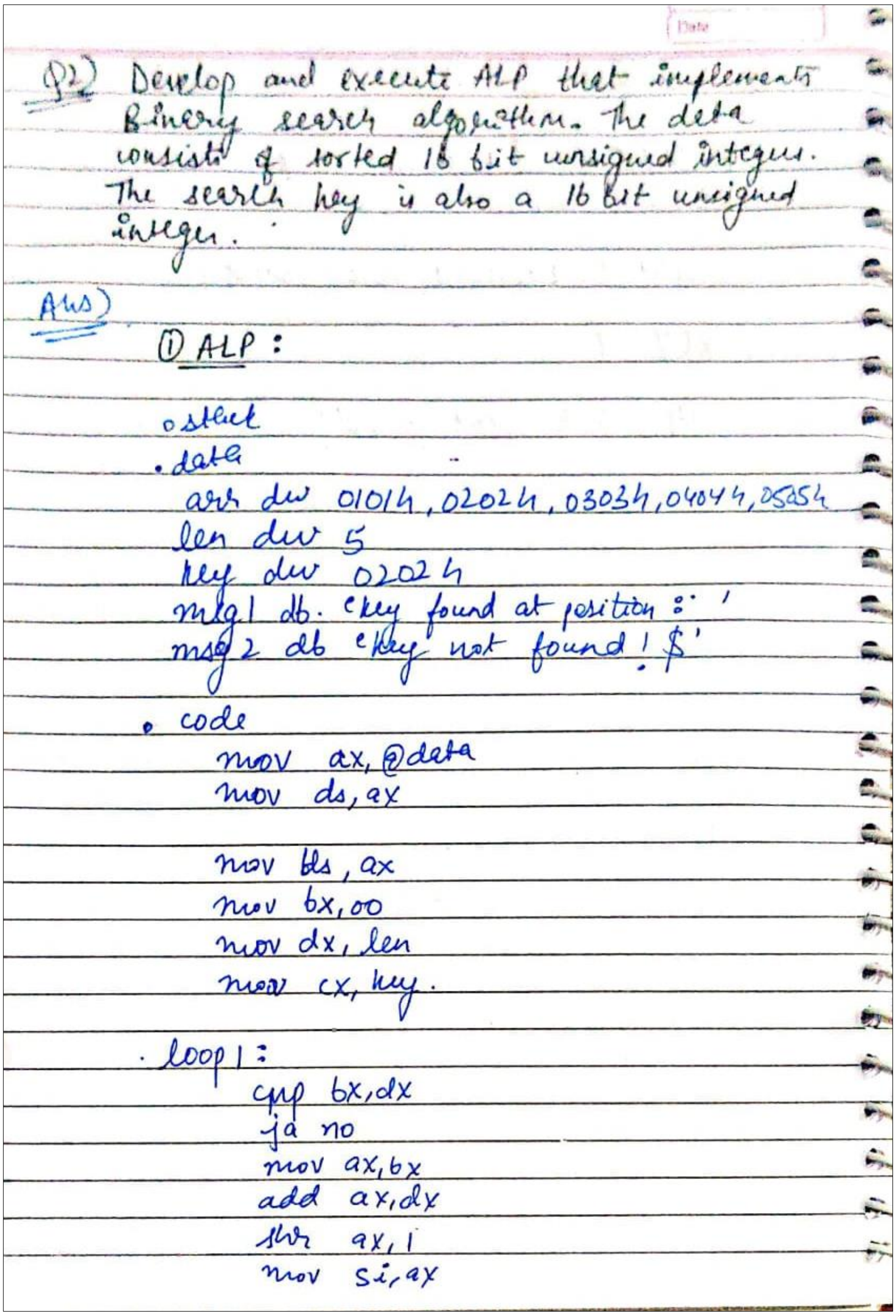
```
F400:0204                F400:0204
F4200: FF 255 RES        BIOS DI
F4201: FF 255 RES        INT 021h
F4202: CD 205 =          IRET
F4203: 21 033 !          ADD [BX + SI], AL
F4204: CF 207 ╧          ADD [BX + SI], AL
F4205: 00 000 NULL       ADD [BX + SI], AL
F4206: 00 000 NULL       ADD [BX + SI], AL
F4207: 00 000 NULL       ADD [BX + SI], AL
F4208: 00 000 NULL       ADD [BX + SI], AL
F4209: 00 000 NULL       ADD [BX + SI], AL
F420A: 00 000 NULL       ADD [BX + SI], AL
F420B: 00 000 NULL       ADD [BX + SI], AL
F420C: 00 000 NULL       ADD [BX + SI], AL
F420D: 00 000 NULL       ADD [BX + SI], AL
F420E: 00 000 NULL       ADD [BX + SI], AL
F420F: 00 000 NULL       ADD [BX + SI], AL
F4210: 00 000 NULL       ADD [BX + SI], AL
F4211: 00 000 NULL       ADD [BX + SI], AL
F4212: 00 000 NULL       ADD [BX + SI], AL
F4213: 00 000 NULL       ADD [BX + SI], AL
F4214: 00 000 NULL       ADD [BX + SI], AL
F4215: 00 000 NULL       ...
```

```
Enter the key: 4 Element is found!
```



```
Enter the key: 4 Element is found!
```

2. Develop and execute ALP that implements Binary search algorithm. The data consists of sorted 16 bit unsigned integers. The search key is also a 16 bit unsigned integer.



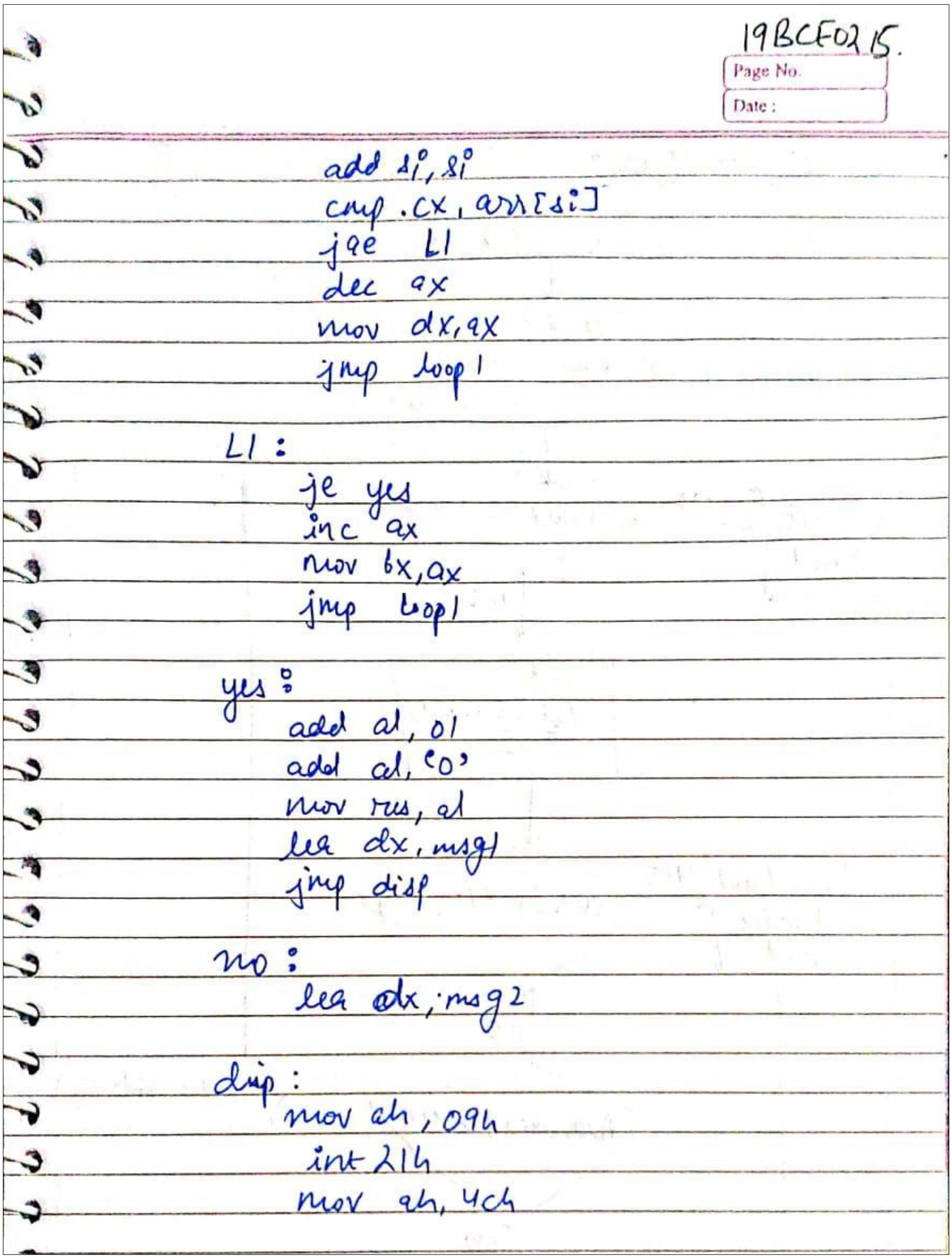
Date

Q2) Develop and execute ALP that implements Binary search algorithm. The data consists of sorted 16 bit unsigned integers. The search key is also a 16 bit unsigned integer.

Ans)

① ALP :

```
.stack
.data
  arr dw 01014,02024,03034,04044,05054
  len dw 5
  key dw 02024
  msg1 db 'key found at position : '
  msg2 db 'key not found!$'

.code
   mov ax,@data
   mov ds,ax

   mov bds,ax
   mov bx,00
   mov dx,len
   mov cx,key.

.loop1:
     cmp bx,dx
     ja no
     mov ax,bx
     add ax,dx
     shr ax,1
     mov si,ax
```

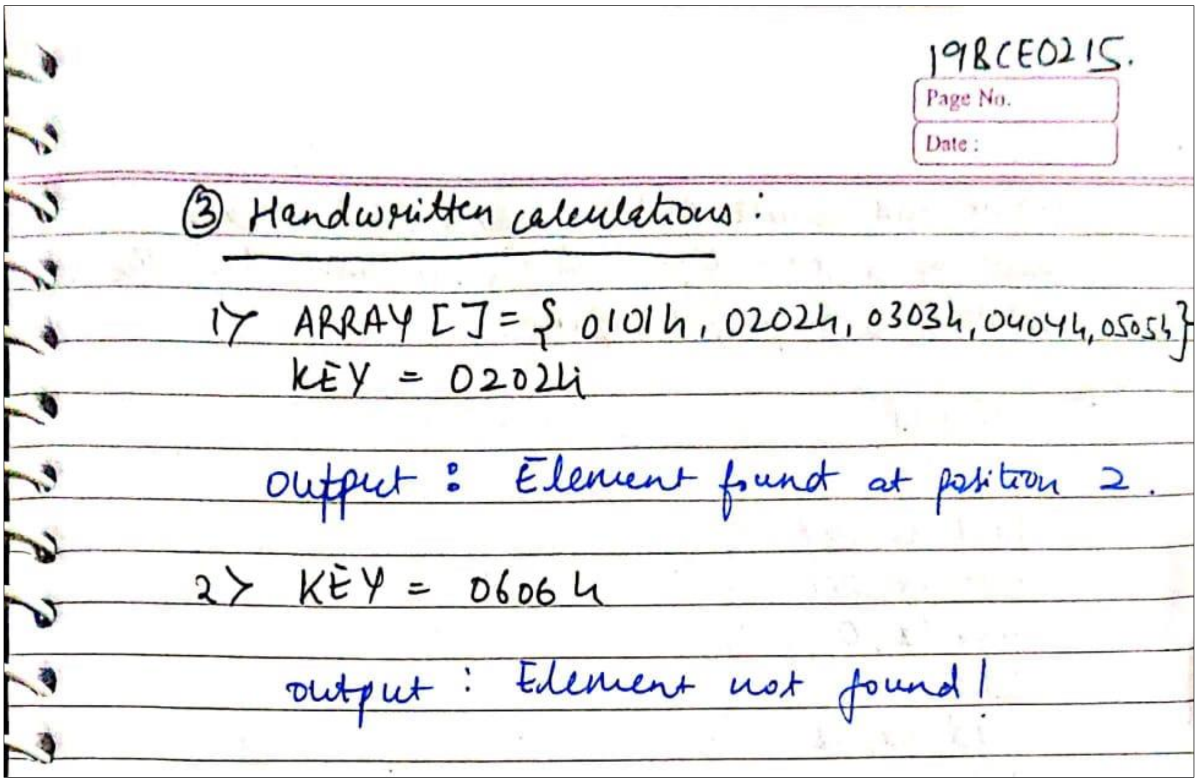


19BCE0215.

Page No.

Date :

```
        add si, si
        cmp .cx, arr[si]
        jge  L1
        dec  ax
        mov  dx,ax
        jmp  loop1

L1 :
       je yes
       inc ax
       mov bx,ax
       jmp  loop1

yes :
      add al, 01
      add al, '0'
      mov res, al
      lea dx, msg1
      jmp disp

no :
      lea dx, msg2

disp :
      mov ah, 09h
       int 21h
      mov ah, 4ch
```



19BCE0215.

Page No.

Date :

③ Handwritten calculations :

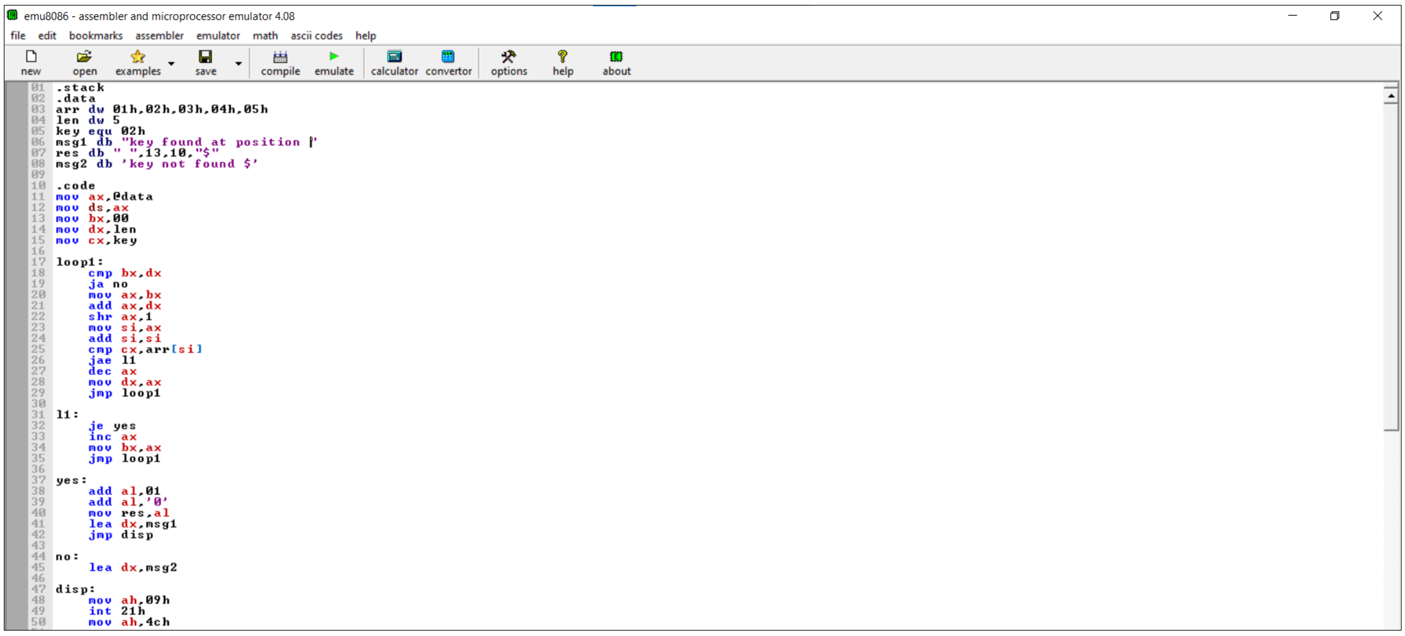
1> ARRAY [ ] = { 0101h, 0202h, 0303h, 0404h, 0505h }
KEY = 0202h

output : Element found at position 2.

2> KEY = 0606h

output : Element not found !

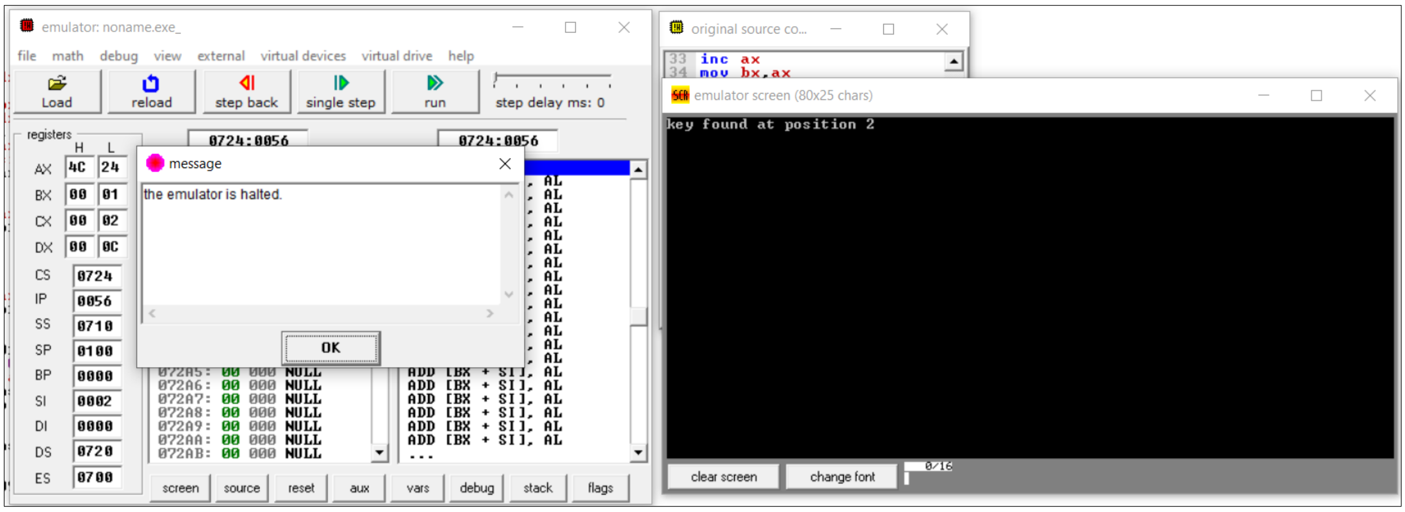
Screenshot of ALP:



```
.stack
.data
arr dw 01h,02h,03h,04h,05h
len dw 5
key equ 02h
msg1 db "key found at position '
res db " ",13,10,"$"
msg2 db 'key not found $'

.code
mov ax,@data
mov ds,ax
mov bx,00
mov dx,len
mov cx,key

loop1:
    cmp bx,dx
    ja no
    mov ax,bx
    add ax,dx
    shr ax,1
    mov si,ax
    add si,si
    cmp cx,arr[si]
    jae l1
    dec ax
    mov dx,ax
    jmp loop1

l1:
    je yes
    inc ax
    mov bx,ax
    jmp loop1

yes:
    add al,01
    add al,'0'
    mov res,al
    lea dx,msg1
    jmp disp

no:
    lea dx,msg2

disp:
    mov ah,09h
    int 21h
    mov ah,4ch
```

emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

Screenshot of Output:



emulator: noname.exe_

```
registers  H  L
AX         4C 24
BX         00 01
CX         00 02
DX         00 0C
CS         0724
IP         0056
SS         0710
SP         0100
BP         0000
SI         0002
DI         0000
DS         0720
ES         0700
```

```
0724:0056                     0724:0056
072A5: 00 000 NULL            ADD [BX + SI], AL
072A6: 00 000 NULL            ADD [BX + SI], AL
072A7: 00 000 NULL            ADD [BX + SI], AL
072A8: 00 000 NULL            ADD [BX + SI], AL
072A9: 00 000 NULL            ADD [BX + SI], AL
072AA: 00 000 NULL            ADD [BX + SI], AL
072AB: 00 000 NULL            ADD [BX + SI], AL
                              ...
```

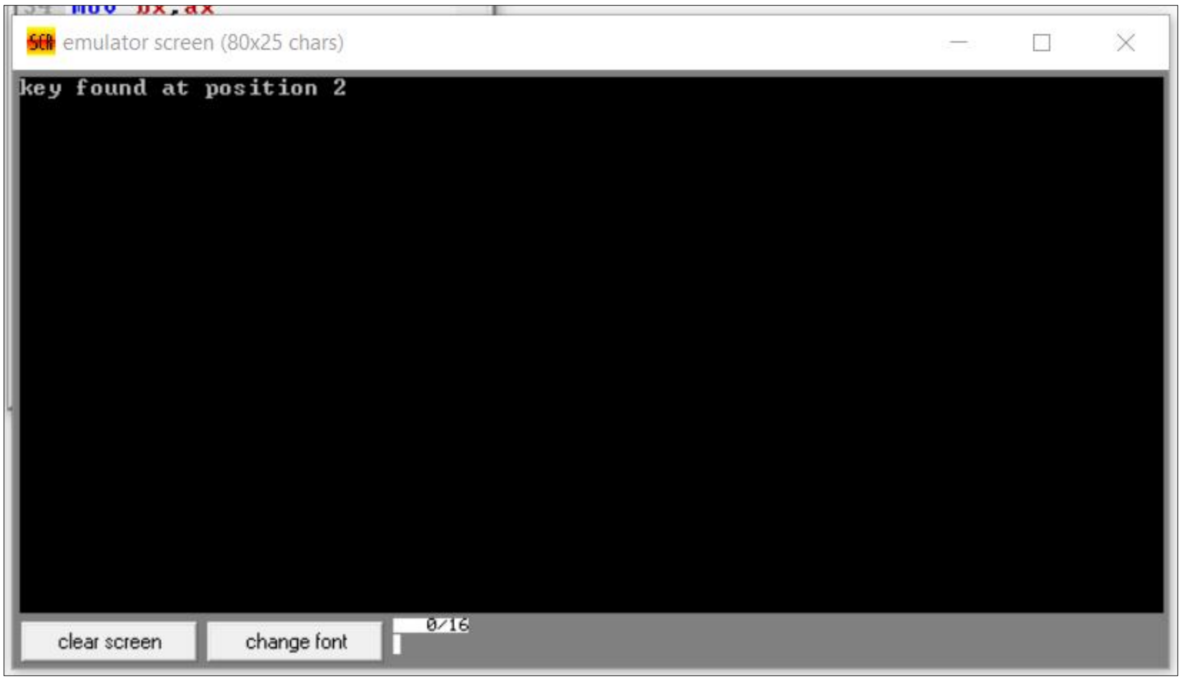
message: the emulator is halted.

original source co...

```
inc ax
mov bx,ax
```

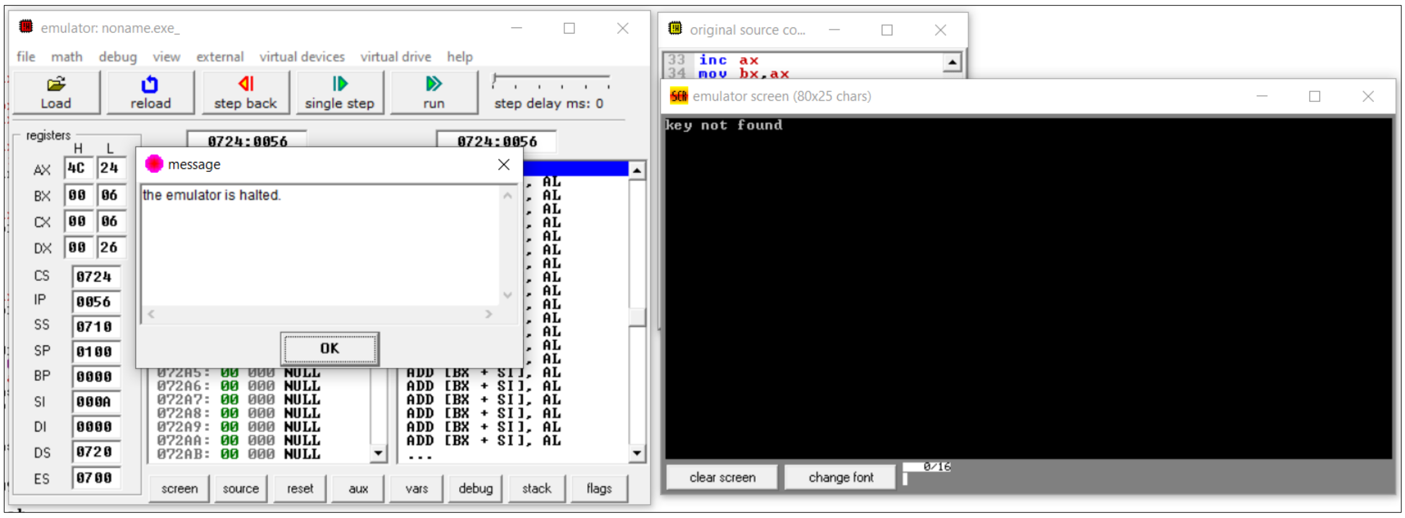
emulator screen (80x25 chars)

```
key found at position 2
```



emulator screen (80x25 chars)

```
key found at position 2
```



emulator: noname.exe_

```
registers  H  L
AX         4C 24
BX         00 06
CX         00 06
DX         00 26
CS         0724
IP         0056
SS         0710
SP         0100
BP         0000
SI         000A
DI         0000
DS         0720
ES         0700
```

```
0724:0056                     0724:0056
072A5: 00 000 NULL            ADD [BX + SI], AL
072A6: 00 000 NULL            ADD [BX + SI], AL
072A7: 00 000 NULL            ADD [BX + SI], AL
072A8: 00 000 NULL            ADD [BX + SI], AL
072A9: 00 000 NULL            ADD [BX + SI], AL
072AA: 00 000 NULL            ADD [BX + SI], AL
072AB: 00 000 NULL            ADD [BX + SI], AL
                              ...
```

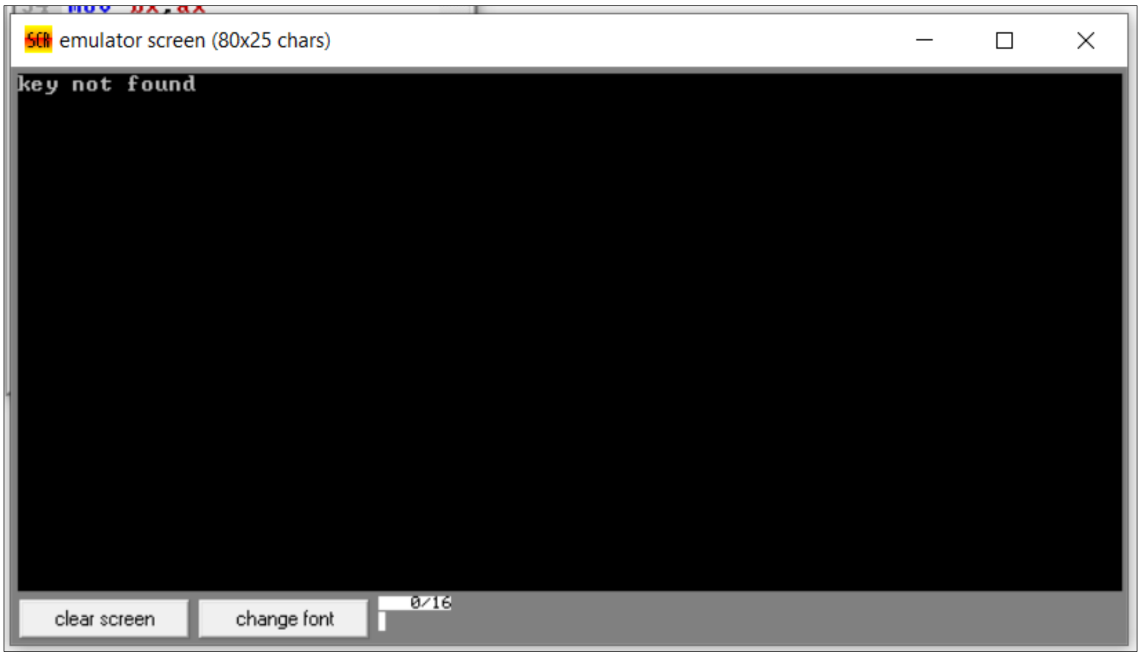
message: the emulator is halted.

original source co...

```
inc ax
mov bx,ax
```

emulator screen (80x25 chars)

```
key not found
```



```
key not found
```