(Question 1)

Develop an ALP to display A-Z and Z-A one at a time.

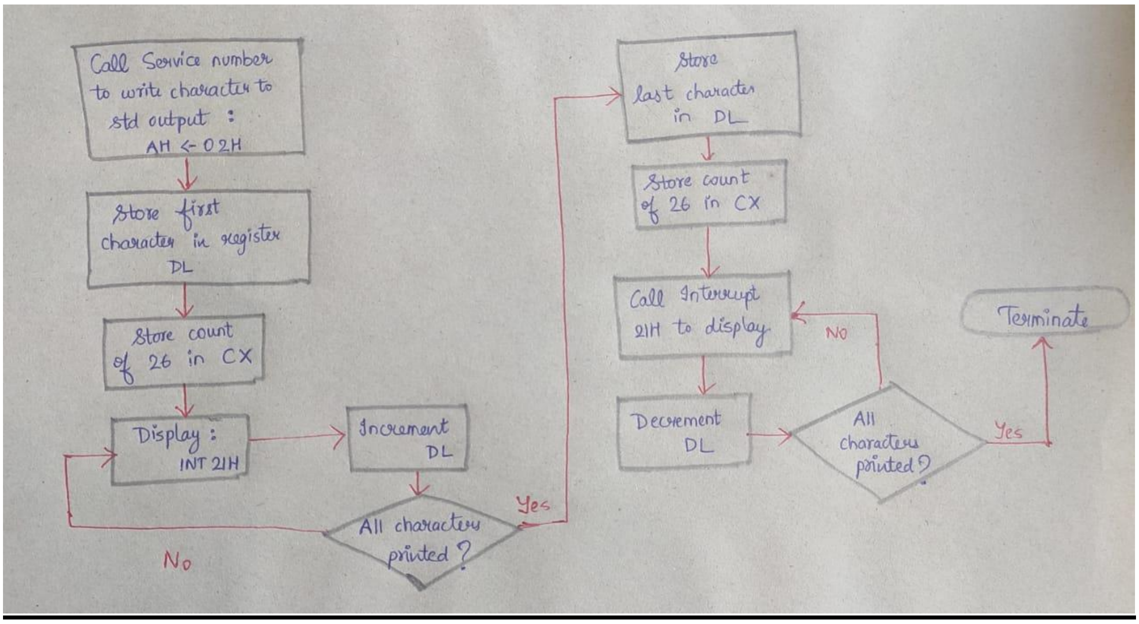
(Aim)

To use string interrupts to display alphabets in sequence.

(Algorithm/Pseudocode)

```
START

Call Service Number 02H to write character to standard output
Move the first character to register DL
Move the count of 26 (alphabets) to counter register CX
Loop to display 26 alphabets in sequence
Call Interrupt 21H to execute
Increment register DL to move forward in sequence
Loop ends

Move the last character to register DL
Move the count of 26 (alphabets) to counter register CX
Loop to display 26 alphabets in reverse sequence
Call Interrupt 21H to execute
Decrement register DL to move forward in sequence
Loop ends

END
```

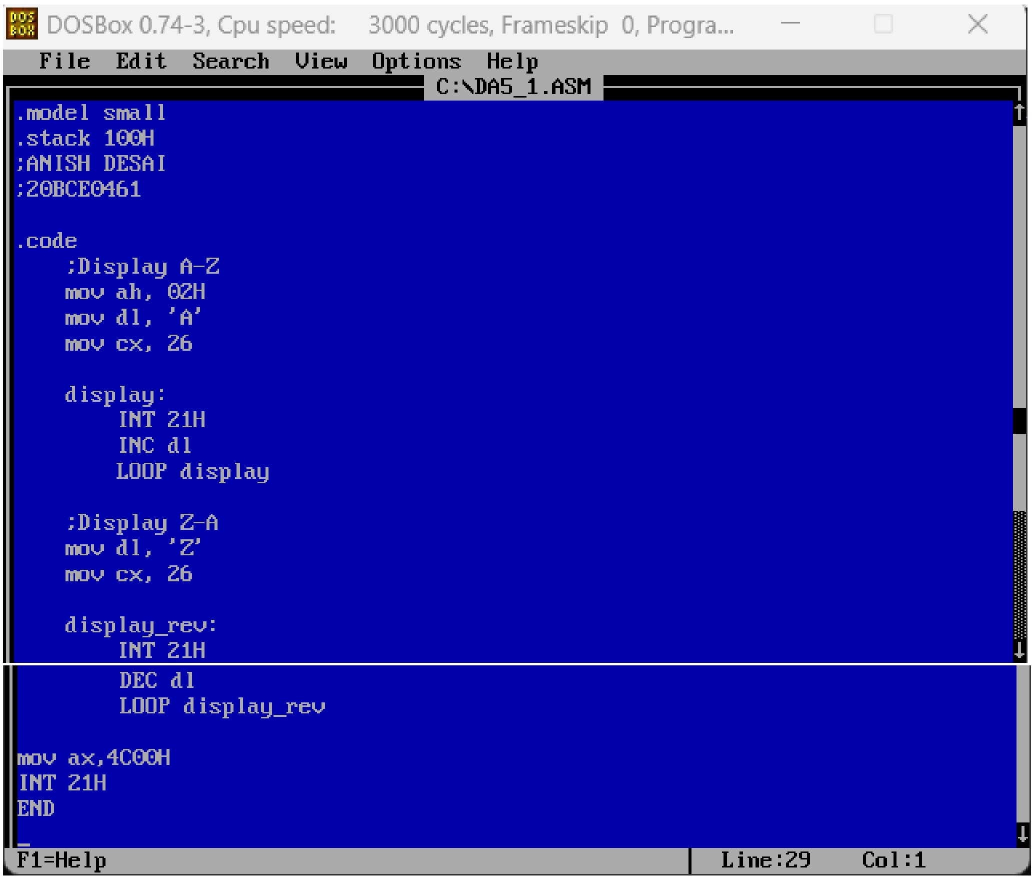
(Flowchart)



Call Service number to write character to std output : AH <- 02H
Store first character in register DL
Store count of 26 in CX
Display : INT 21H
Increment DL
All characters printed ?
Yes
No
Store last character in DL
Store count of 26 in CX
Call Interrupt 21H to display
No
Decrement DL
All characters printed ?
Yes
Terminate

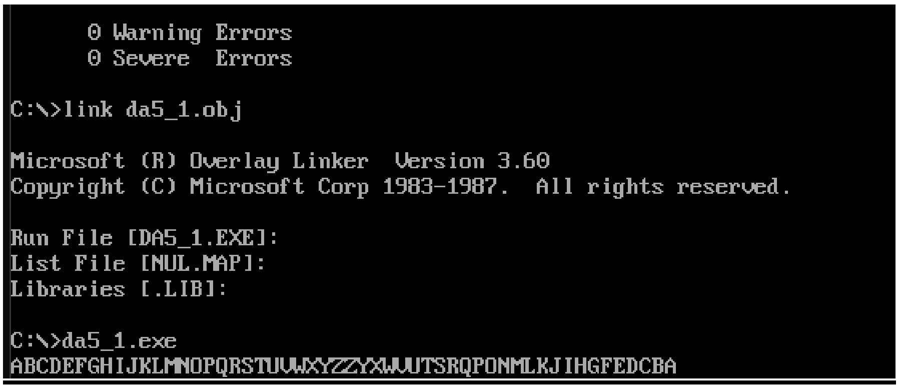
(ALP Code)

```
.model small
.stack 100H
;ANISH DESAI
;20BCE0461

.code
   ;Display A-Z
   mov ah, 02H
   mov dl, 'A'
   mov cx, 26

   display:
      INT 21H
      INC dl
      LOOP display

   ;Display Z-A
   mov dl, 'Z'
   mov cx, 26

   display_rev:
      INT 21H
      DEC dl
      LOOP display_rev

mov ax,4C00H
INT 21H
END
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\DA5_1.ASM
F1=Help
Line:29 Col:1

```
.model small
.stack 100H
;ANISH DESAI
;20BCE0461

.code
    ;Display A-Z
    mov ah, 02H
    mov dl, 'A'
    mov cx, 26

    display:
        INT 21H
        INC dl
        LOOP display

    ;Display Z-A
    mov dl, 'Z'
    mov cx, 26

    display_rev:
        INT 21H
        DEC dl
        LOOP display_rev

mov ax,4C00H
INT 21H
END
```

(MASM Output)



```
      0 Warning Errors
      0 Severe  Errors

C:\>link da5_1.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [DA5_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>da5_1.exe
ABCDEFGHIJKLMNOPQRSTUVWXYZZYXWVUTSRQPONMLKJIHGFEDCBA
```

(Result)

As we can see from above MASM output screen, the alphabets have been successfully printed in sequence and reverse sequence one at a time using Loop instructions.

(Question 2)

Develop an ALP to reverse a string.

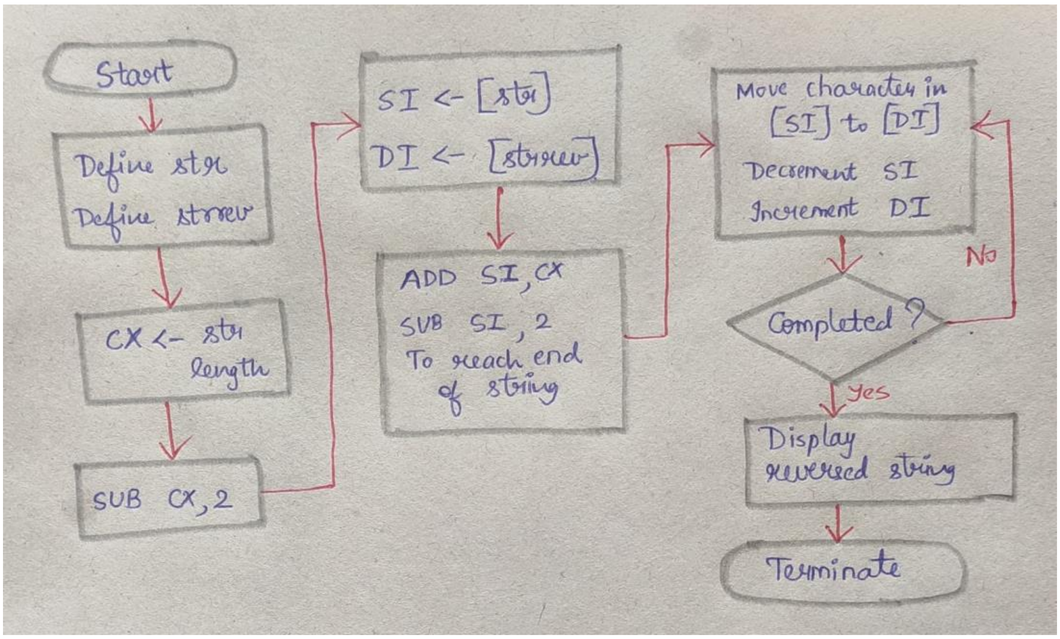
(Aim)

To reverse any given string.

(Algorithm/Pseudocode)

```
START

Define a string in str
Define an empty string in strrev to store reversed string
Assign length to register CX and subtract 2
Assign str address to SI
Assign strrev address to DI
Add string length to SI and subtract 2 to reach the end of str

LOOP
    Move character in address SI to address DI
    Decrement SI to traverse backwards
    Increment DI to store incrementally
END LOOP

Display reversed string strrev

END
```

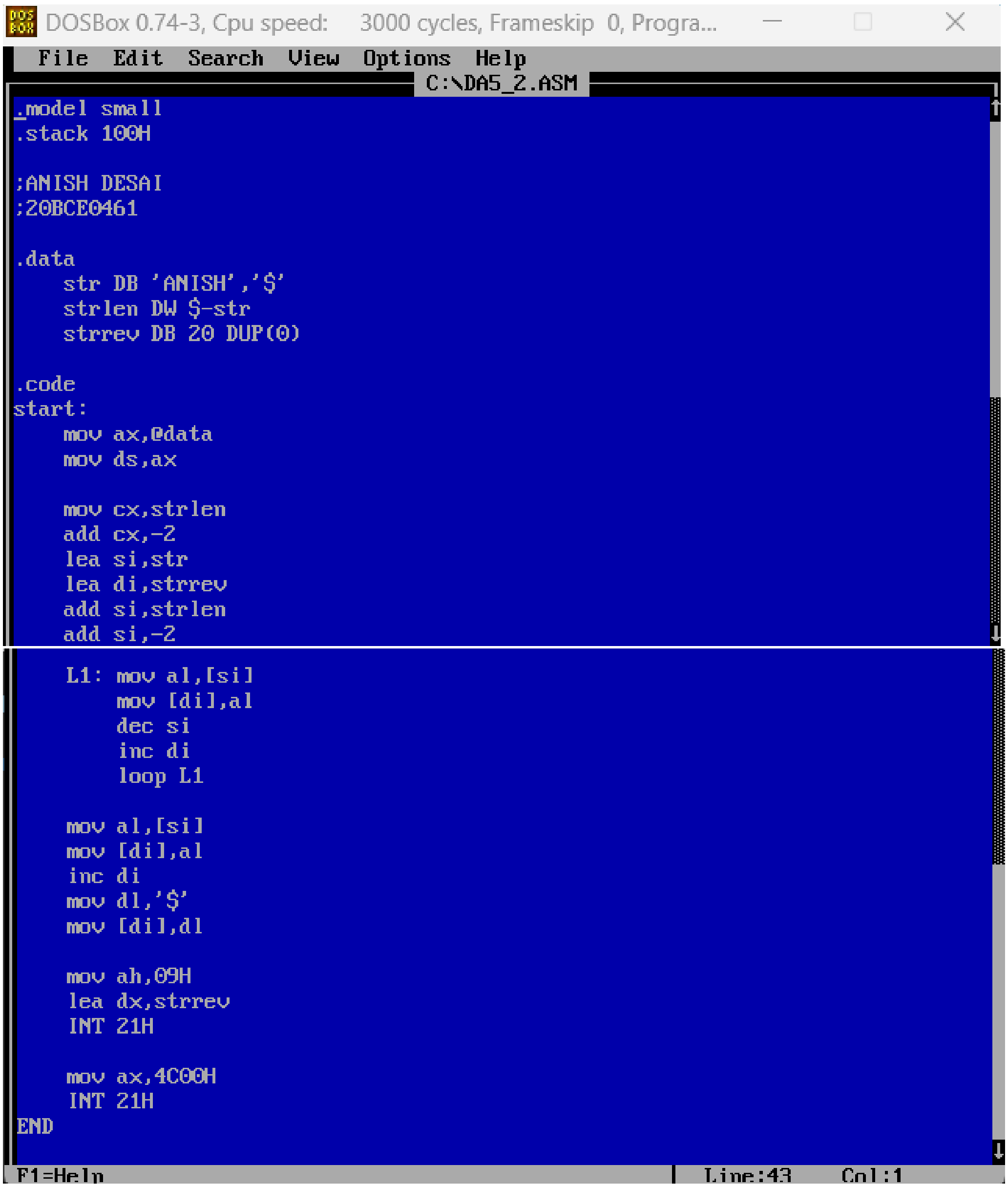
(Flowchart)



(ALP Code)

```
.model small
.stack 100H

;ANISH DESAI
;20BCE0461

.data
   str DB 'ANISH','$'
   strlen DW $-str
   strrev DB 20 DUP(0)

.code
start:
   mov ax,@data
   mov ds,ax

   mov cx,strlen
   add cx,-2
   lea si,str
   lea di,strrev
   add si,strlen
   add si,-2

   L1: mov al,[si]
      mov [di],al
      dec si
      inc di
      loop L1

   mov al,[si]
   mov [di],al
   inc di
   mov dl,'$'
   mov [di],dl

   mov ah,09H
```

```
    lea dx,strrev
    INT 21H

    mov ax,4C00H
    INT 21H
END
```

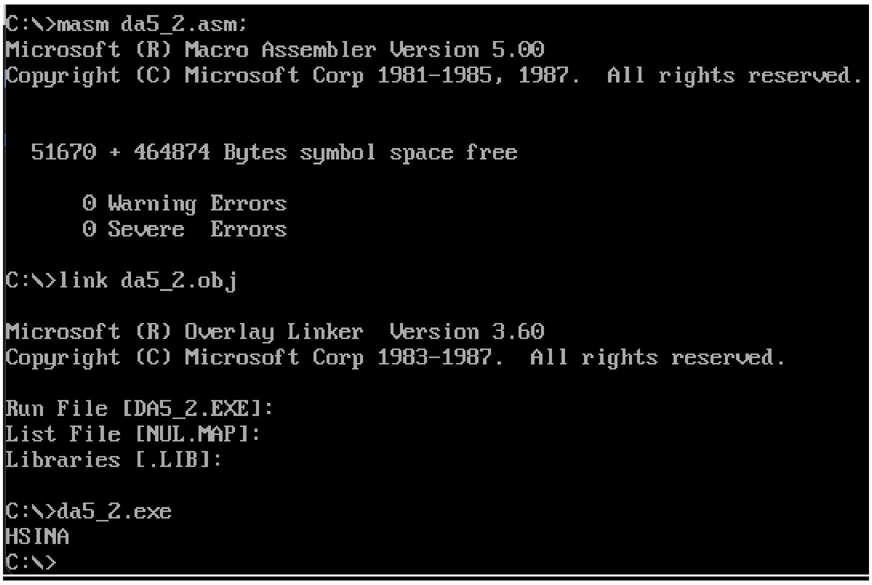


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\DA5_2.ASM

```
.model small
.stack 100H

;ANISH DESAI
;20BCE0461

.data
    str DB 'ANISH','$'
    strlen DW $-str
    strrev DB 20 DUP(0)

.code
start:
    mov ax,@data
    mov ds,ax

    mov cx,strlen
    add cx,-2
    lea si,str
    lea di,strrev
    add si,strlen
    add si,-2

    L1: mov al,[si]
        mov [di],al
        dec si
        inc di
        loop L1

    mov al,[si]
    mov [di],al
    inc di
    mov dl,'$'
    mov [di],dl

    mov ah,09H
    lea dx,strrev
    INT 21H

    mov ax,4C00H
    INT 21H
END
```

F1=Help
Line:43   Col:1

(MASM Output)



```
C:\>masm da5_2.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.


  51670 + 464874 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link da5_2.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [DA5_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>da5_2.exe
HSINA
C:\>
```

(Result)

As we can see from the above MASM output screen, the reversed string of the string stored has been successfully displayed upon execution.

(Question 3)

Extend the above ALP to check for palindrome.

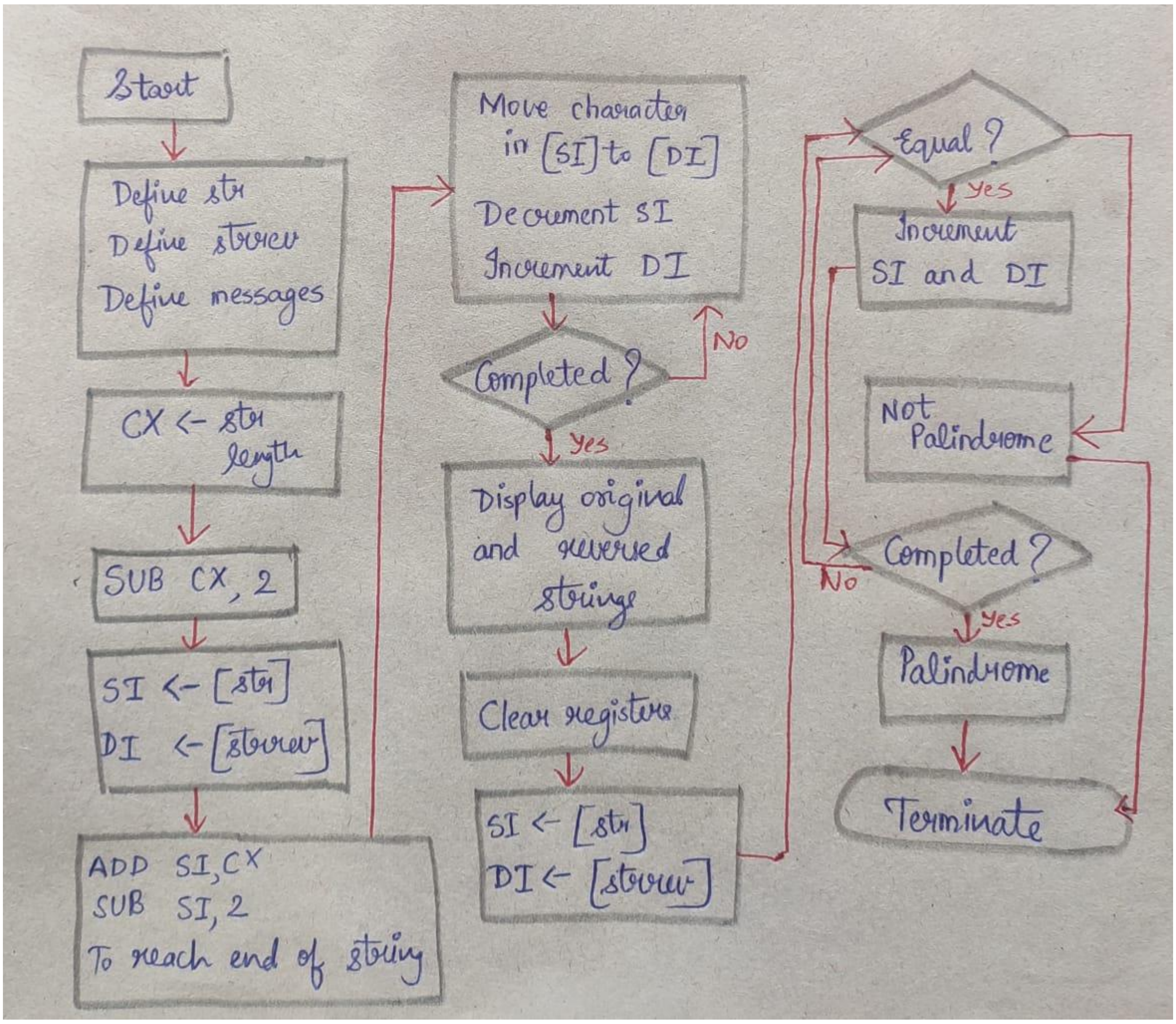
(Aim)

To check for palindrome string by first reversing the given string and then comparing each letter of both the strings.

(Algorithm/Pseudocode)

```
START

Define the messages to be displayed
Define a string in str
Define an empty string in strrev to store reversed string
Assign length to register CX and subtract 2
Assign str address to SI
Assign strrev address to DI
Add string length to SI and subtract 2 to reach the end of str

LOOP
    Move character in address SI to address DI
    Decrement SI to traverse backwards
    Increment DI to store incrementally
END LOOP

Display original string and reversed string

Clear registers
Assign str address to SI
Assign strrev address to DI
Loop to compare each letter in SI and DI incrementally
IF letters not equal
Jump to NotPALindrome
ELSE, continue to loop till end reached
Display Palindrome message
IN NotPALindrome, display Not Palindrome message
END
```

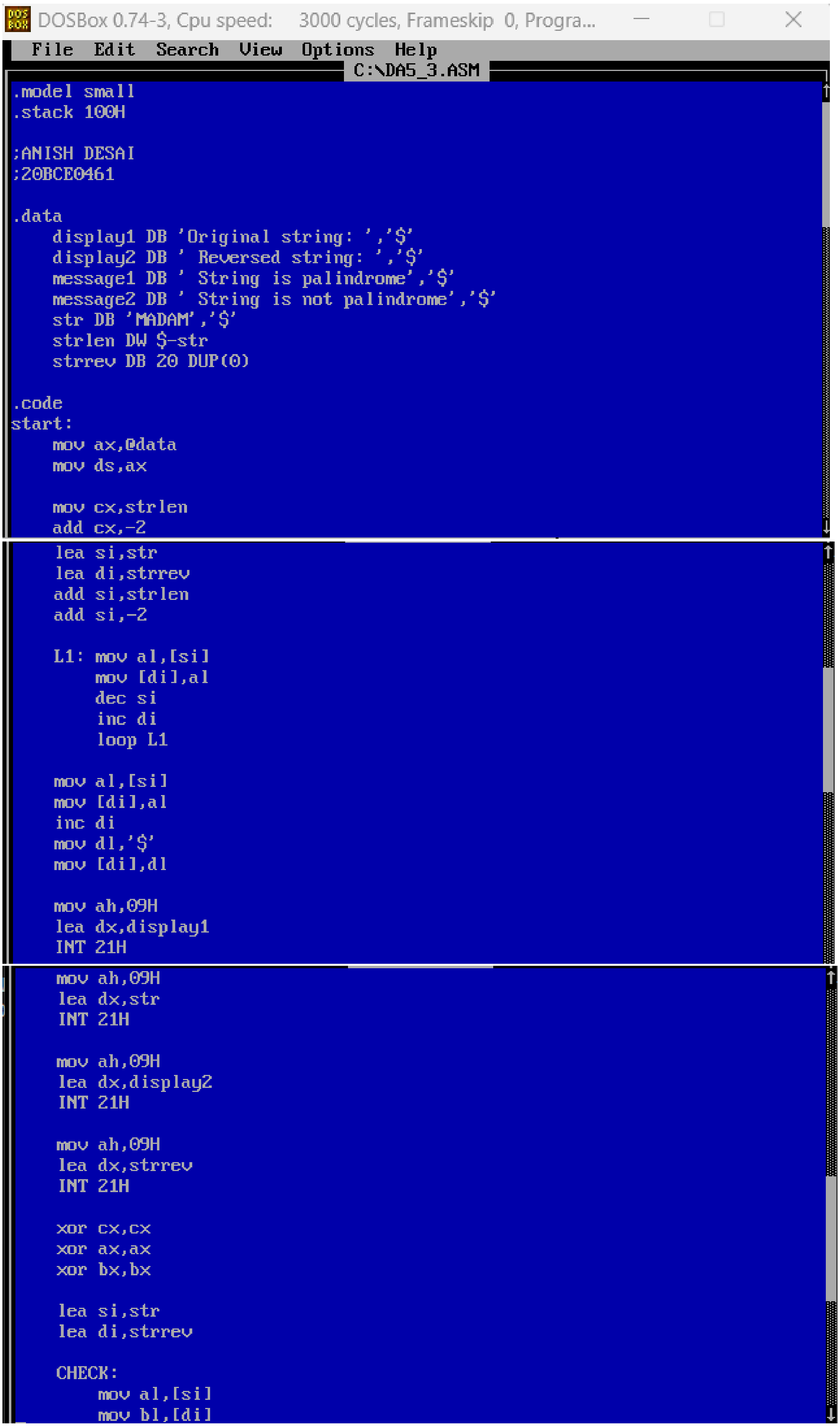
(Flowchart)



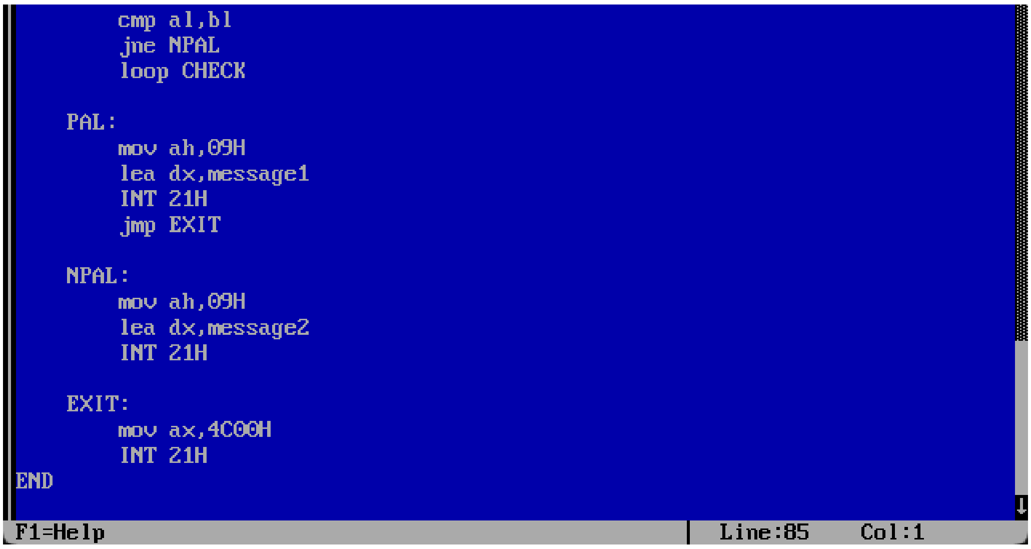
(ALP Code)

```
.model small
.stack 100H

;ANISH DESAI
;20BCE0461

.data
   display1 DB 'Original string: ','$'
   display2 DB ' Reversed string: ','$'
   message1 DB ' String is palindrome','$'
   message2 DB ' String is not palindrome','$'
   str DB 'MADAM','$'
   strlen DW $-str
```

```
    strrev DB 20 DUP(0)

.code
start:
    mov ax,@data
    mov ds,ax

    mov cx,strlen
    add cx,-2
    lea si,str
    lea di,strrev
    add si,strlen
    add si,-2

    L1: mov al,[si]
        mov [di],al
        dec si
        inc di
        loop L1

    mov al,[si]
    mov [di],al
    inc di
    mov dl,'$'
    mov [di],dl

    mov ah,09H
    lea dx,display1
    INT 21H

    mov ah,09H
    lea dx,str
    INT 21H

    mov ah,09H
    lea dx,display2
    INT 21H
    mov ah,09H
    lea dx,strrev
    INT 21H

    xor cx,cx
    xor ax,ax
    xor bx,bx

    lea si,str
    lea di,strrev

    CHECK:
        mov al,[si]
        mov bl,[di]
        cmp al,bl
        jne NPAL
        loop CHECK

    PAL:
        mov ah,09H
        lea dx,message1
        INT 21H
        jmp EXIT

    NPAL:
        mov ah,09H
        lea dx,message2
        INT 21H

    EXIT:
        mov ax,4C00H
        INT 21H
END
```

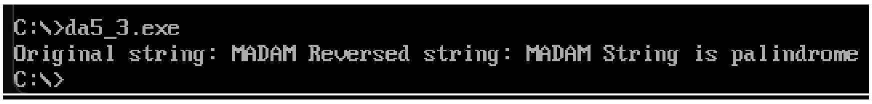

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\DA5_3.ASM

```
.model small
.stack 100H

;ANISH DESAI
;20BCE0461

.data
    display1 DB 'Original string: ','$'
    display2 DB ' Reversed string: ','$'
    message1 DB ' String is palindrome','$'
    message2 DB ' String is not palindrome','$'
    str DB 'MADAM','$'
    strlen DW $-str
    strrev DB 20 DUP(0)

.code
start:
    mov ax,@data
    mov ds,ax

    mov cx,strlen
    add cx,-2
    lea si,str
    lea di,strrev
    add si,strlen
    add si,-2

    L1: mov al,[si]
        mov [di],al
        dec si
        inc di
        loop L1

    mov al,[si]
    mov [di],al
    inc di
    mov dl,'$'
    mov [di],dl

    mov ah,09H
    lea dx,display1
    INT 21H
    mov ah,09H
    lea dx,str
    INT 21H

    mov ah,09H
    lea dx,display2
    INT 21H

    mov ah,09H
    lea dx,strrev
    INT 21H

    xor cx,cx
    xor ax,ax
    xor bx,bx

    lea si,str
    lea di,strrev

    CHECK:
        mov al,[si]
        mov bl,[di]
```

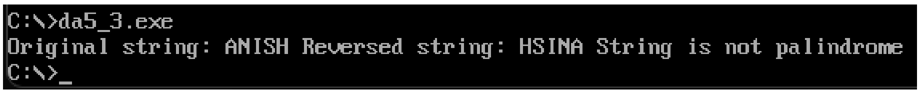


```
        cmp al,bl
        jne NPAL
        loop CHECK

    PAL:
        mov ah,09H
        lea dx,message1
        INT 21H
        jmp EXIT

    NPAL:
        mov ah,09H
        lea dx,message2
        INT 21H

    EXIT:
        mov ax,4C00H
        INT 21H
END
```

(MASM Output)



```
C:\>da5_3.exe
Original string: MADAM Reversed string: MADAM String is palindrome
C:\>
```



```
C:\>da5_3.exe
Original string: ANISH Reversed string: HSINA String is not palindrome
C:\>_
```

(Result)

As we can see in the above MASM output screen, for the word ‘MADAM’, the string is reversed and **palindrome check successful**. And also, for the word ‘ANISH’, the string is reversed and **not palindrome check successful**.

(Question 4)

Develop an ALP to find the smallest number in an array of unsorted numbers. The smallest number should be displayed using interrupts.

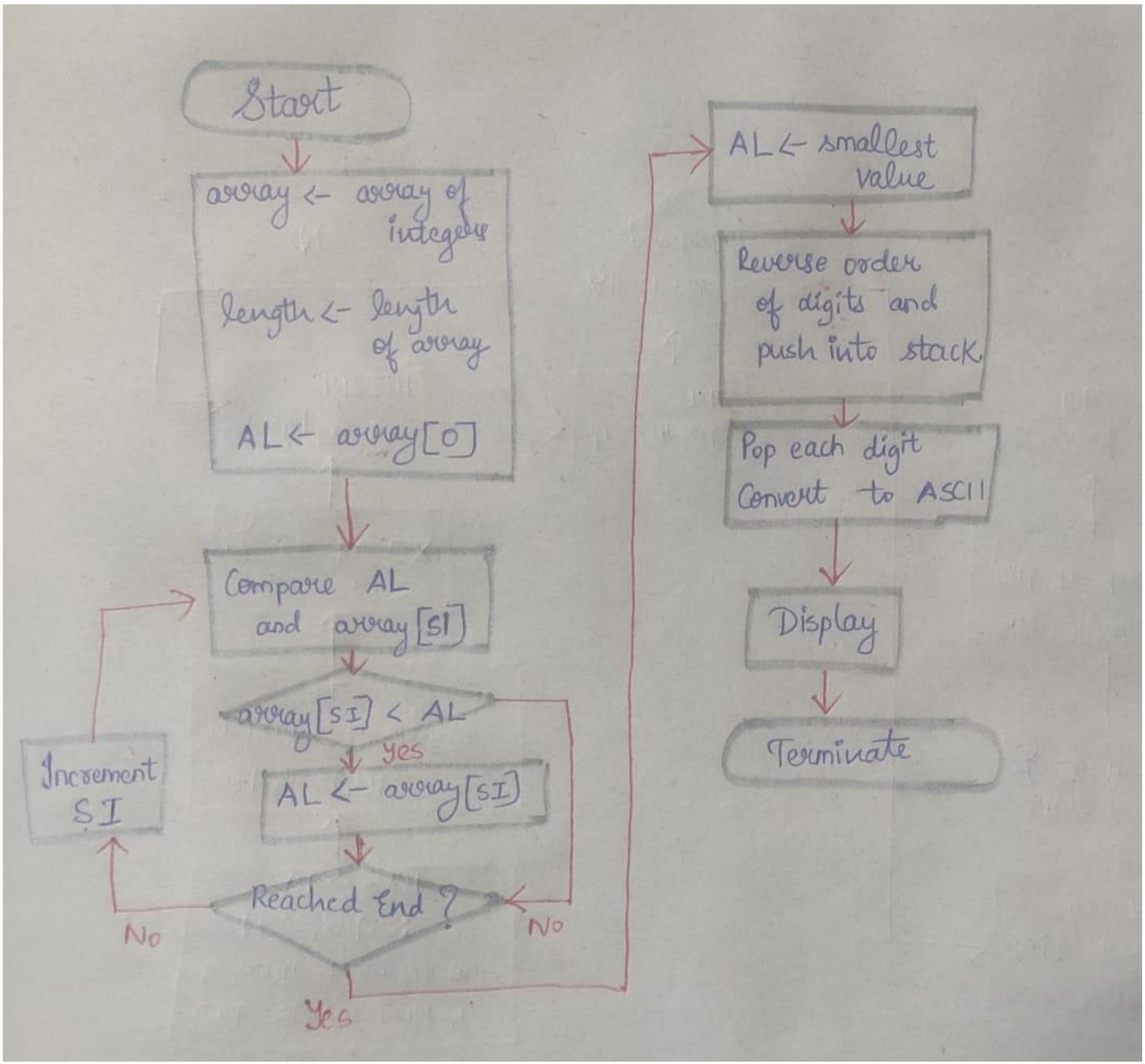
(Aim)

To find the smallest number in an unsorted array and display it using interrupts.

(Algorithm/Pseudocode)

```
START

Assign integers to ‘array’ and length of array to ‘length’
Assign number of digits pushed onto stack to ‘pushed’
Load first element in register AL

LOOP
Compare current value in AL with each value
IF less, the pointed value = new minimum
END LOOP

AL ← Smallest value
Reverse the order of digits in number
Pop each digit out of stack and convert into ASCII character
Display

END
```

(Flowchart)

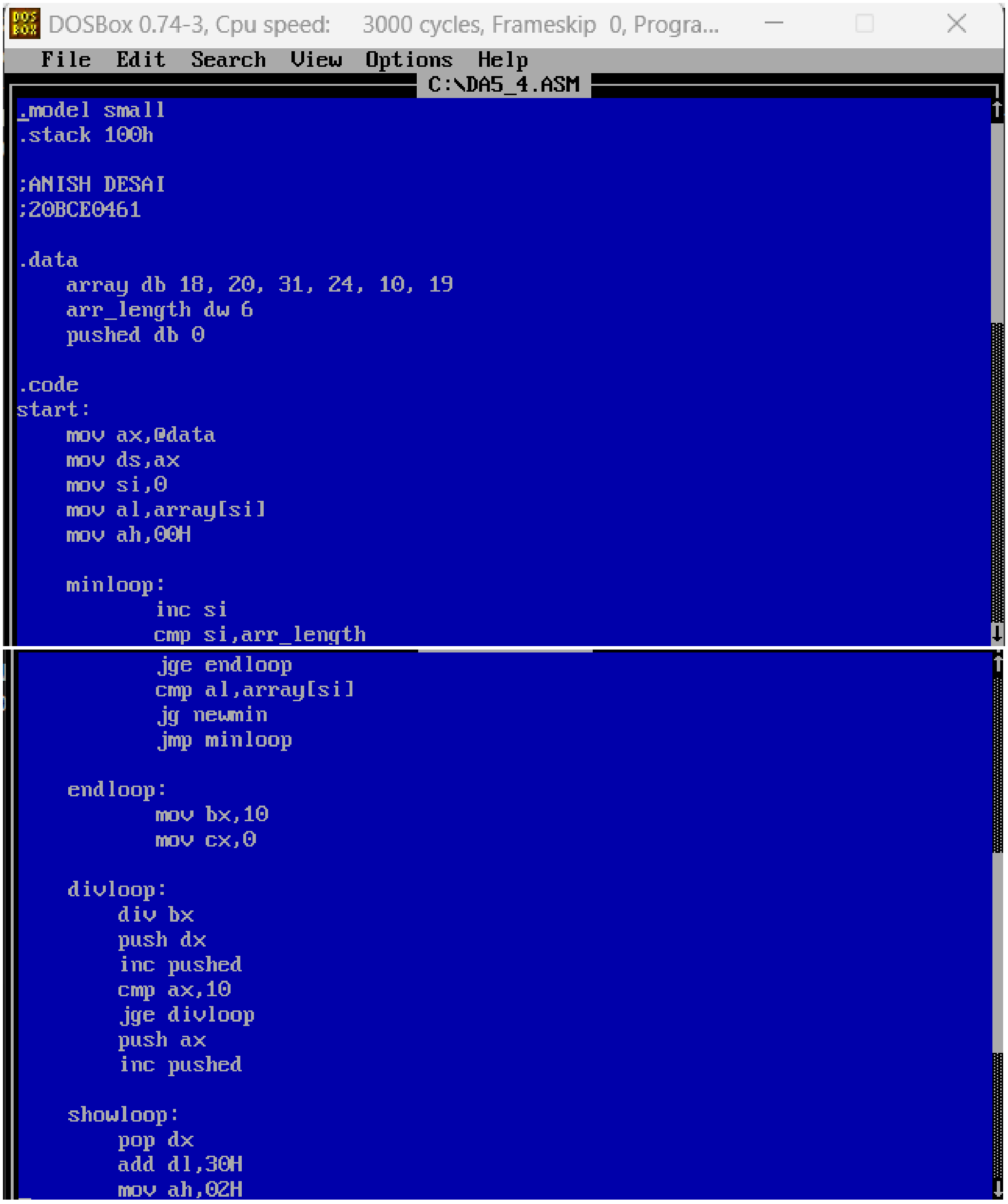


(ALP Code)

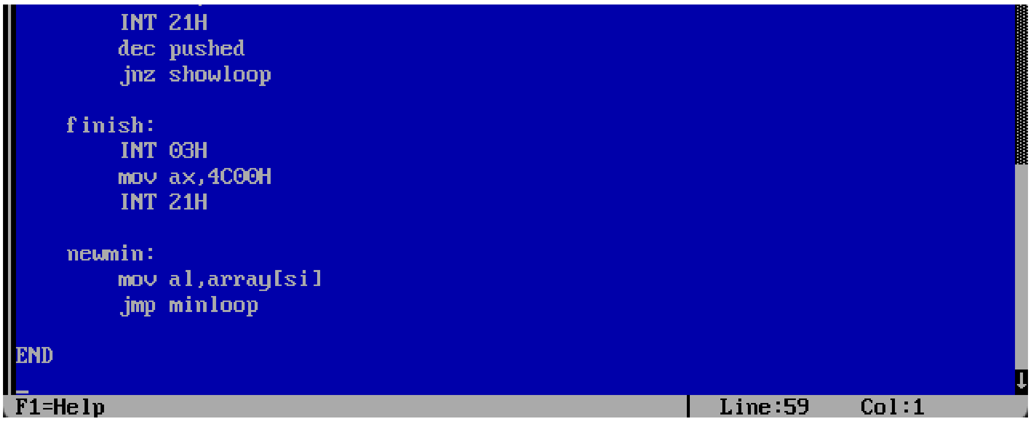
```
.model small
.stack 100h

;ANISH DESAI
;20BCE0461

.data
   array db 18, 20, 31, 24, 10, 19
   arr_length dw 6
   pushed db 0
```

```
.code
start:
    mov ax,@data
    mov ds,ax
    mov si,0
    mov al,array[si]
    mov ah,00H

    minloop:
               inc si
              cmp si,arr_length
              jge endloop
              cmp al,array[si]
              jg newmin
              jmp minloop

    endloop:
              mov bx,10
              mov cx,0

    divloop:
        div bx
        push dx
        inc pushed
        cmp ax,10
        jge divloop
        push ax
        inc pushed

    showloop:
        pop dx
        add dl,30H
        mov ah,02H
        INT 21H
        dec pushed
        jnz showloop
```

```
  finish:
     INT 03H
     mov ax,4C00H
     INT 21H

  newmin:
     mov al,array[si]
     jmp minloop

END
```

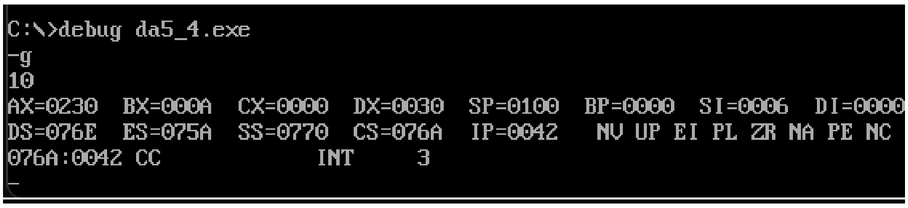

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\DA5_4.ASM

```
.model small
.stack 100h

;ANISH DESAI
;20BCE0461

.data
    array db 18, 20, 31, 24, 10, 19
    arr_length dw 6
    pushed db 0

.code
start:
    mov ax,@data
    mov ds,ax
    mov si,0
    mov al,array[si]
    mov ah,00H

    minloop:
           inc si
           cmp si,arr_length
           jge endloop
           cmp al,array[si]
           jg newmin
           jmp minloop

    endloop:
           mov bx,10
           mov cx,0

    divloop:
        div bx
        push dx
        inc pushed
        cmp ax,10
        jge divloop
        push ax
        inc pushed

    showloop:
        pop dx
        add dl,30H
        mov ah,02H
```



```
        INT 21H
        dec pushed
        jnz showloop

    finish:
        INT 03H
        mov ax,4C00H
        INT 21H

    newmin:
        mov al,array[si]
        jmp minloop

END
```

(MASM Output)



```
C:\>debug da5_4.exe
-g
10
AX=0230  BX=000A  CX=0000  DX=0030  SP=0100  BP=0000  SI=0006  DI=0000
DS=076E  ES=075A  SS=0770  CS=076A  IP=0042   NV UP EI PL ZR NA PE NC
076A:0042 CC            INT     3
-
```

(Result)

As we can see from above MASM output screen, the smallest number from the defined array has been found and successfully displayed.