MIT LAB 11

SAHIL BONDRE: U18CO021

1. Write ALP to ADD/SUB ‘n’ 16 bit numbers stored in consecutive memory location

```
.model small

.data
  sum dw ?
  carry db ?
  nums dw 0011h, 0022h, 0033h, 0044h, 0055h

.code
  mov ax,@data
  mov ds,ax

  mov ax, 0000H
  mov cx, 05H ; counter
  mov bl, 00H ; to count carry
  mov si, offset nums ; init si to start of nums
  up:
  add ax, [si]
  jnc skip
  inc bl ;
  skip:
  inc si
  inc si
  loop up
  mov sum, ax
  mov carry, bl

  mov ah, 4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q1.EXE
-u
076A:0000 B86C07        MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 B80000        MOV     AX,0000
076A:0008 B90500        MOV     CX,0005
076A:000B B300          MOV     BL,00
076A:000D BE0900        MOV     SI,0009
076A:0010 0304          ADD     AX,[SI]
076A:0012 7302          JNB     0016
076A:0014 FEC3          INC     BL
076A:0016 46            INC     SI
076A:0017 46            INC     SI
076A:0018 E2F6          LOOP    0010
076A:001A A30600        MOV     [0006],AX
076A:001D 881E0800      MOV     [0008],BL
-_
```

```
Program terminated normally
-d 076c:0000
076C:0000  00 B4 4C CD 21 00 FF 00-00 11 00 22 00 33 00 44   ..L.!......".3.D
076C:0010  00 55 00 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   .U..r...z...BX..
076C:0020  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076C:0030  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076C:0040  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076C:0050  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076C:0060  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
076C:0070  00 AB B0 00 AB AB AA 8B-1E 54 04 A1 D8 03 40 A3   .........T....@.
-_
```

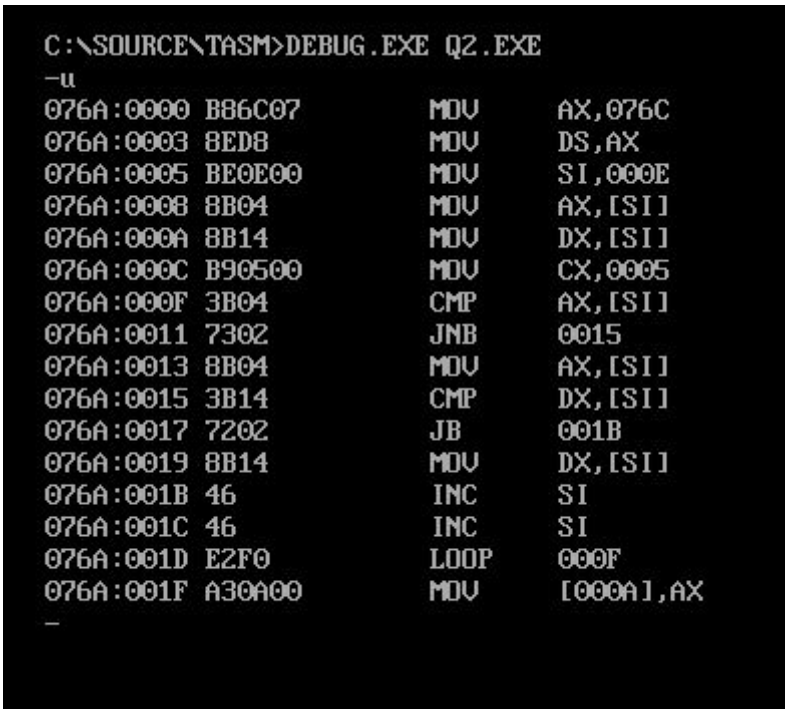
2. Write a Program to find smallest/largest number in a given array of 16 bits numbers.

```
.model small

.data
  max dw ?
  min dw ?
  nums dw 1111H, 0EEFFH, 0EFFFH, 2222H, 9999H

.code
  mov ax,@data
  mov ds,ax
  mov si, offset nums ; init si to start of nums
  mov ax, [si]
  mov dx, [si]
  mov cx, 05H  ;counter
  up:
  cmp ax, [si]
```

```
  jnc next
  mov ax, [si]
  next:
  cmp dx, [si]
  jc skip
  mov dx, [si]
  skip:
  inc si
  inc si
  loop up
  mov max, ax
  mov min, dx
  mov ah, 4ch
  int 21h
end
```



```
076A:0040 8C0E5034      MOV     [3450],CS
-d 076c:0000
076C:0000  0A 00 89 16 0C 00 B4 4C-CD 21 FF EF 11 11 11 11   .......L.!......
076C:0010  FF EE FF EF 22 22 99 99-7A 04 04 BE 42 58 B4 00   ....""..z...BX..
076C:0020  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076C:0030  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076C:0040  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076C:0050  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076C:0060  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
076C:0070  00 AB B0 00 AB AB AA 8B-1E 54 04 A1 D8 03 40 A3   .........T....@.
-_
```

3. Write a Program to sort 16 bits given numbers in ascending /descending order.

```
.model small

.data
  nums dw 0048H, 0012H, 0111H, 0FFFFh, 9999h

.code
  mov ax, @data
  mov ds, ax
  mov ax, 0000h
  mov bx, 0005h
  up1:
  mov si, offset nums
  mov cx, 0005h
  up:
  mov dx, [si]
  inc si
  inc si
  mov ax, [si]
  cmp ax, dx
  jnc skip
  xchg [si], dx
  xchg [si - 2], ax
  skip:
  loop up
  dec bx
  jnz up1
  mov ah, 4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q3.EXE
-u
076A:0000 B86C07        MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 B80000        MOV     AX,0000
076A:0008 BB0500        MOV     BX,0005
076A:000B BE0A00        MOV     SI,000A
076A:000E B90500        MOV     CX,0005
076A:0011 8B14          MOV     DX,[SI]
076A:0013 46            INC     SI
076A:0014 46            INC     SI
076A:0015 8B04          MOV     AX,[SI]
076A:0017 3BC2          CMP     AX,DX
076A:0019 7305          JNB     0020
076A:001B 8714          XCHG    DX,[SI]
076A:001D 8744FE        XCHG    AX,[SI-02]
-
```

```
Program terminated normally
-d 076c:0000
076C:0000  E2 EF 4B 75 E6 B4 4C CD-21 00 12 00 48 00 11 01   ..Ku..L.!...H...
076C:0010  72 05 99 99 FF FF 83 0E-7A 04 04 BE 42 58 B4 00   r.......z...BX..
076C:0020  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076C:0030  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076C:0040  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076C:0050  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076C:0060  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
076C:0070  00 AB B0 00 AB AB AA 8B-1E 54 04 A1 D8 03 40 A3   .........T....@.
-
```

4. Write a Program to find occurrences of a given number in a list of N numbers given through keyboard.

```
.model small

.8086
.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter data: $"
  msg1 db cr,lf,"Enter a search charecter:$"
  msg2 db cr,lf,"Frequency: $"
  c db ?
  st1 db 80 dup('$')
  cnt db 0h

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm
```

```
.code
  mov ax,@data
  mov ds,ax
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  mov ah,01h
  int 21h
  mov c,al
  mov si,offset st1
  u:
  mov bl,[si]
  cmp c,bl
  jz down
  inc si
  loop u
  jmp e
  down:
  inc cnt
  inc si
  loop u
  e:
  print msg2
  mov ah,02h
  mov dh,00h
  mov dl,cnt
  add dl,30h
  int 21h
  mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q4.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0E00        MOV     DX,000E
076A:000A CD21          INT     21
076A:000C BE4600        MOV     SI,0046
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter data: sahilbondre@gmail.com

Enter a search charecter:a
Frequency: 2
Program terminated normally
-_
```

5. Write a Program to move a string from source to destination.

```
.model small
.8086

.data
  p1 db "sahil$"
  p2 db 6 dup(0)

.code
  mov ax,@data
  mov ds,ax

  mov si,offset p1
  mov di,offset p2
  mov cx,0006h
  up:
  mov al,[si]
  mov [di],al
  inc si
  inc di
  dec cx
  jnz up
  mov ax,4c00h
  int 21h
```

```
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q5.EXE
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 BE0C00        MOV     SI,000C
076A:0008 BF1200        MOV     DI,0012
076A:000B B90600        MOV     CX,0006
076A:000E 8A04          MOV     AL,[SI]
076A:0010 8805          MOV     [DI],AL
076A:0012 46            INC     SI
076A:0013 47            INC     DI
076A:0014 49            DEC     CX
076A:0015 75F7          JNZ     000E
076A:0017 B8004C        MOV     AX,4C00
076A:001A CD21          INT     21
076A:001C 7361          JNB     007F
076A:001E 68            DB      68
076A:001F 69            DB      69
-_
```

```
Program terminated normally
-d 076b:0000
076B:0000  88 05 46 47 49 75 F7 B8-00 4C CD 21 73 61 68 69   ..FGIu...L.!sahi
076B:0010  6C 24 73 61 68 69 6C 24-00 8C 0E 50 34 E8 6C 35   l$sahil$...P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
-
```

6. Write a Program to reverse a given string.

```
model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter string: $"
  msg1 db cr,lf, "String: $"
  msg2 db cr,lf,"Reverse: $"
  st1 db 80 dup('$')
  st2 db 80 dup('$')
```

```
  print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm

.code
  mov ax,@data
  mov ds,ax
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  print st1
  mov si,offset st1
  add si,cx
  dec si
  mov di,offset st2
  m:
  mov bl,[si]
  mov [di],bl
  dec si
  inc di
  dec cx
  jnz m
  print msg2
  print st2
  mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q6.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0200        MOV     DX,0002
076A:000A CD21          INT     21
076A:000C BE2800        MOV     SI,0028
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter string: sahil

String: sahil
Reverse: lihas
Program terminated normally
-
```

7. Write a Program to perform case conversion (U to L, L to U) for a given string.

To uppercase:

```
model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter a string: $"
  msg1 db cr,lf, "String: $"
  msg2 db cr,lf,"Uppercase: $"
  st1 db 80 dup('$')
  st2 db 80 dup('$')

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm

.code
  mov ax,@data
  mov ds,ax
```

```
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  print st1
  mov si,offset st1
  mov di,offset st2
  u:
  mov al,[si]
  cmp al,'a'
  jb next
  cmp al,'z'
  ja next
  sub al,20h
  next: mov [di],al
  inc si
  inc di
  loop u
  print msg2
  print st2

  mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q7a.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0800        MOV     DX,0008
076A:000A CD21          INT     21
076A:000C BE3200        MOV     SI,0032
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter a string: sahil

String: sahil
Uppercase: SAHIL
Program terminated normally
-_
```

```
.model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter a string: $"
  msg1 db cr,lf, "String: $"
  msg2 db cr,lf,"Lowecase:$"
  st1 db 80 dup('$')
  st2 db 80 dup('$')
  print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
  endm

.code
  mov ax,@data
  mov ds,ax
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
```

```
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  print st1
  mov si,offset st1
  mov di,offset st2
  u:
  mov al,[si]
  cmp al,'A'
  jb next
  cmp al,'Z'
  ja next
  add al,20h
  next: mov [di],al
  inc si
  inc di
  loop u
  print msg2
  print st2
  mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q7b.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0800        MOV     DX,0008
076A:000A CD21          INT     21
076A:000C BE3000        MOV     SI,0030
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter a string: SAHIL

String: SAHIL
Lowecase:sahil
Program terminated normally
-_
```

8. Write a Program to merge two strings entered through keyboard.

```
model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "First String: $"
  msg1 db cr,lf,"Second String: $"
  msg2 db cr,lf,"Concat: $"
  st1 db 80 dup('$')
  st2 db 80 dup('$')
  st3 db 80 dup('$')

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm

.code
  mov ax,@data
  mov ds,ax
  print msg
```

```
mov si,offset st1
mov cx,0000h
up:
mov ah,01h
int 21h
cmp al,0dh
je stp
mov [si],al
inc si
inc cx
jmp up
stp:
print msg1
mov si,offset st2
mov dh,00h
u:
mov ah,01h
int 21h
cmp al,0dh
je s
mov [si],al
inc si
inc dh
jmp u
s:
mov si,offset st1
mov di,offset st3
me:
mov dl,[si]
mov [di],dl
inc si
inc di
loop me
mov si,offset st2
go:
mov dl,[si]
mov [di],dl
dec cx
inc si
inc di
dec dh
jnz go
print msg2
print st3
mov ah,4ch
int 21h
```

```
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q8.EXE
-u
076A:0000 B87007        MOV     AX,0770
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0A00        MOV     DX,000A
076A:000A CD21          INT     21
076A:000C BE3600        MOV     SI,0036
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
First String: Hello

Second String: World

Concat: HelloWorld
Program terminated normally
-
```

9. Write a Program to search a character in a given string

```
model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter String: $"
  msg1 db cr,lf,"Enter Search Charecter: $"
  msg2 db cr,lf,"Found!$"
  msg3 db cr,lf,"Not Found$"
  c db ?
  st1 db 80 dup('$')

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
```

```
endm

.code
  mov ax,@data
  mov ds,ax
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  mov ah,01h
  int 21h
  mov c,al
  mov si,offset st1
  u:
  mov bl,[si]
  cmp c,bl
  jz down
  inc si
  loop u
  print msg3
  jmp e
  down:
  print msg2
  e: mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q9.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0200        MOV     DX,0002
076A:000A CD21          INT     21
076A:000C BE4200        MOV     SI,0042
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter String: sahil

Enter Search Charecter: s
Found!
Program terminated normally
-
-_
```

10. Write a Program to find occurrences of a given character in a given string through keyboard.

```
.model small

.8086
.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter data: $"
  msg1 db cr,lf,"Enter a search charecter:$"
  msg2 db cr,lf,"Frequency: $"
  c db ?
  st1 db 80 dup('$')
  cnt db 0h

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm

.code
```

```
  mov ax,@data
  mov ds,ax
  print msg
  mov si,offset st1
  mov cx,0000h
  up:
  mov ah,01h
  int 21h
  cmp al,0dh
  je stp
  mov [si],al
  inc si
  inc cx
  jmp up
  stp:
  print msg1
  mov ah,01h
  int 21h
  mov c,al
  mov si,offset st1
  u:
  mov bl,[si]
  cmp c,bl
  jz down
  inc si
  loop u
  jmp e
  down:
  inc cnt
  inc si
  loop u
  e:
  print msg2
  mov ah,02h
  mov dh,00h
  mov dl,cnt
  add dl,30h
  int 21h
  mov ah,4ch
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q4.EXE
-u
076A:0000 B86F07        MOV     AX,076F
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0E00        MOV     DX,000E
076A:000A CD21          INT     21
076A:000C BE4600        MOV     SI,0046
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter data: sahilbondre@gmail.com

Enter a search charecter:a
Frequency: 2
Program terminated normally
-_
```

11. Program to check whether given substring exist in a main string or not?

```
model small
.8086

.data
  cr equ 0dh
  lf equ 0ah
  msg db "Enter String: $"
  msg1 db cr,lf,"Enter Substring: $"
  msg2 db cr,lf,"Found!$"
  msg3 db cr,lf,"Not Found$"
  st1 db 80 dup('$')
  st2 db 80 dup('$')

print macro msg
  mov ah,09h
  mov dx,offset msg
  int 21h
endm

.code
  mov ax,@data
  mov ds,ax
  print msg
```

```
mov si,offset st1
mov cx,0000h
up:
mov ah,01h
int 21h
cmp al,0dh
je stp
mov [si],al
inc si
inc cx
jmp up
stp:
print msg1
mov si,offset st2
mov bx,0000h
u:
mov ah,01h
int 21h
cmp al,0dh
je stop
mov [si],al
inc si
inc bx
jmp u
stop:
mov si,offset st1
mov di,offset st2
cmp cx,bx
jc down
mov dx,0000h
go:
mov al,[si]
cmp al,[di]
je me
mov dx,0000h
mov di,offset st2
inc si
loop go
jmp en
me:
inc dx
inc si
inc di
cmp dx,bx
jnz go
en:
```

```
  cmp dx,bx
  jnz down
  print msg2
  jmp e
  down:
  print msg3
  e: mov ah,4ch
  int 21h
end
```

```
-u
076A:0000 B87107        MOV     AX,0771
076A:0003 8ED8          MOV     DS,AX
076A:0005 B409          MOV     AH,09
076A:0007 BA0A00        MOV     DX,000A
076A:000A CD21          INT     21
076A:000C BE4200        MOV     SI,0042
076A:000F B90000        MOV     CX,0000
076A:0012 B401          MOV     AH,01
076A:0014 CD21          INT     21
076A:0016 3C0D          CMP     AL,0D
076A:0018 7406          JZ      0020
076A:001A 8804          MOV     [SI],AL
076A:001C 46            INC     SI
076A:001D 41            INC     CX
076A:001E EBF2          JMP     0012
-g
Enter String: sahil

Enter Substring: ah

Found!
Program terminated normally
-_
```