KRISHNA PANDEY **-** U20CS110 **- MIT ASSIGNMENT - 09**

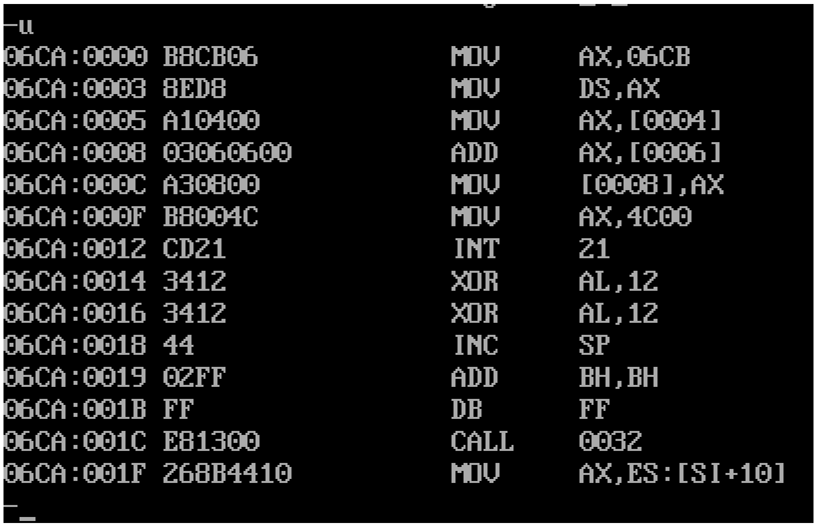
1. Write an assembly language program in 8086, to find the

    - Addition of two 16 bit numbers

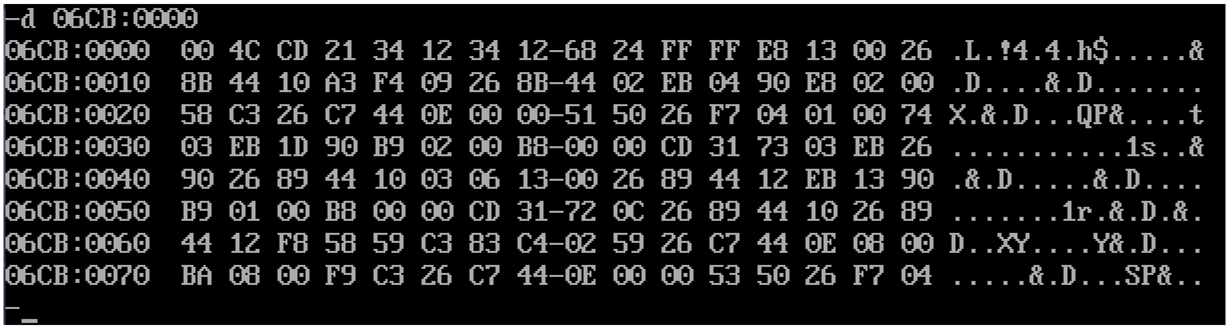
Code:

```
.8086
.model small

.data
    n1 dw 1234h
    n2 dw 1234h
    n3 dw ?

.code
    mov ax, @data
    mov ds, ax

    mov ax, n1
    add ax, n2
    mov n3, ax

    mov ax, 4c00h
    int 21h
    end
```

Output:



```
-u
06CA:0000 B8CB06            MOV     AX,06CB
06CA:0003 8ED8              MOV     DS,AX
06CA:0005 A10400            MOV     AX,[0004]
06CA:0008 03060600          ADD     AX,[0006]
06CA:000C A30800            MOV     [0008],AX
06CA:000F B8004C            MOV     AX,4C00
06CA:0012 CD21              INT     21
06CA:0014 3412              XOR     AL,12
06CA:0016 3412              XOR     AL,12
06CA:0018 44                INC     SP
06CA:0019 02FF              ADD     BH,BH
06CA:001B FF                DB      FF
06CA:001C E81300            CALL    0032
06CA:001F 268B4410          MOV     AX,ES:[SI+10]
-_
```



```
-d 06CB:0000
06CB:0000  00 4C CD 21 34 12 34 12-68 24 FF FF E8 13 00 26 .L.!4.4.h$.....&
06CB:0010  8B 44 10 A3 F4 09 26 8B-44 02 EB 04 90 E8 02 00 .D....&.D.......
06CB:0020  58 C3 26 C7 44 0E 00 00-51 50 26 F7 04 01 00 74 X.&.D...QP&....t
06CB:0030  03 EB 1D 90 B9 02 00 B8-00 00 CD 31 73 03 EB 26 ...........1s..&
06CB:0040  90 26 89 44 10 03 06 13-00 26 89 44 12 EB 13 90 .&.D.....&.D....
06CB:0050  B9 01 00 B8 00 00 CD 31-72 0C 26 89 44 10 26 89 .......1r.&.D.&.
06CB:0060  44 12 F8 58 59 C3 83 C4-02 59 26 C7 44 0E 08 00 D..XY....Y&.D...
06CB:0070  BA 08 00 F9 C3 26 C7 44-0E 00 00 53 50 26 F7 04 .....&.D...SP&..
-_
```

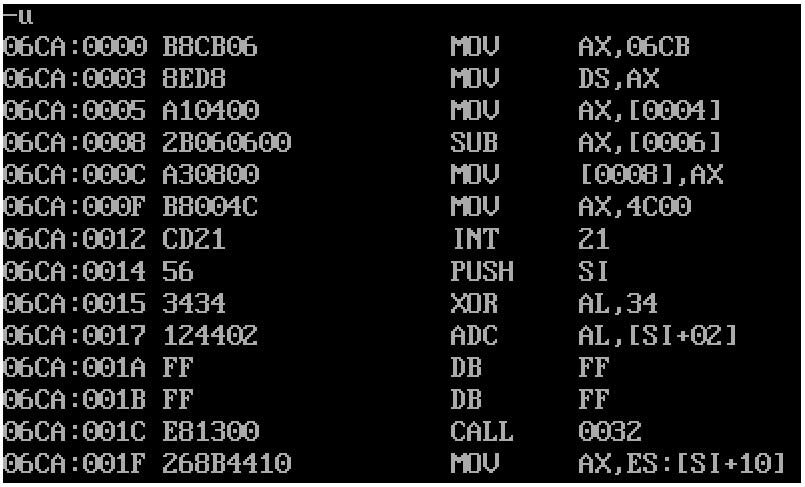
- Subtraction of two 16 bit numbers

Code:

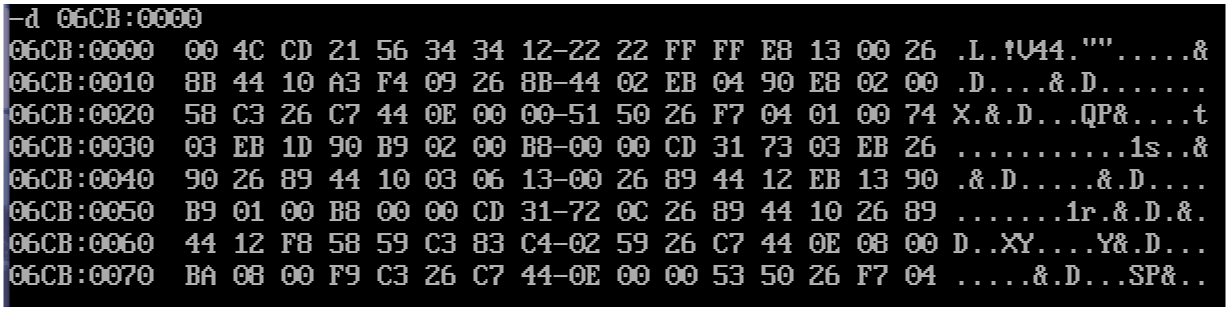
```
.8086
.model small

.data
   n1 dw 3456h
   n2 dw 1234h
   n3 dw ?

.code
   mov ax, @data
   mov ds, ax

   mov ax, n1
   sub ax, n2
```

```
mov n3, ax

mov ax, 4c00h
int 21h
end
```

Output:



```
-u
06CA:0000 B8CB06            MOV     AX,06CB
06CA:0003 8ED8              MOV     DS,AX
06CA:0005 A10400            MOV     AX,[0004]
06CA:0008 2B060600          SUB     AX,[0006]
06CA:000C A30800            MOV     [0008],AX
06CA:000F B8004C            MOV     AX,4C00
06CA:0012 CD21              INT     21
06CA:0014 56                PUSH    SI
06CA:0015 3434              XOR     AL,34
06CA:0017 124402            ADC     AL,[SI+02]
06CA:001A FF                DB      FF
06CA:001B FF                DB      FF
06CA:001C E81300            CALL    0032
06CA:001F 268B4410          MOV     AX,ES:[SI+10]
```



```
-d 06CB:0000
06CB:0000  00 4C CD 21 56 34 34 12-22 22 FF FF E8 13 00 26  .L.!V44."".....&
06CB:0010  8B 44 10 A3 F4 09 26 8B-44 02 EB 04 90 E8 02 00  .D....&.D.......
06CB:0020  58 C3 26 C7 44 0E 00 00-51 50 26 F7 04 01 00 74  X.&.D...QP&....t
06CB:0030  03 EB 1D 90 B9 02 00 B8-00 00 CD 31 73 03 EB 26  ...........1s..&
06CB:0040  90 26 89 44 10 03 06 13-00 26 89 44 12 EB 13 90  .&.D.....&.D....
06CB:0050  B9 01 00 B8 00 00 CD 31-72 0C 26 89 44 10 26 89  .......1r.&.D.&.
06CB:0060  44 12 F8 58 59 C3 83 C4-02 59 26 C7 44 0E 08 00  D..XY....Y&.D...
06CB:0070  BA 08 00 F9 C3 26 C7 44-0E 00 00 53 50 26 F7 04  .....&.D...SP&..
```

2. Write an assembly language program in 8086, to find the

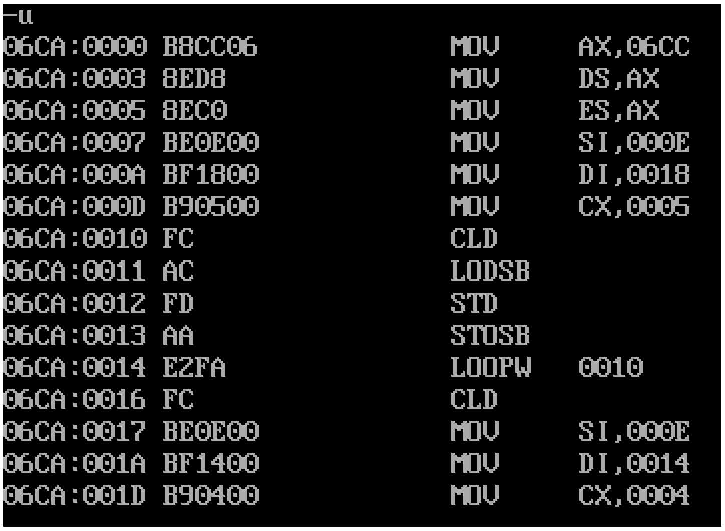
- String is palindrome or not

Code:

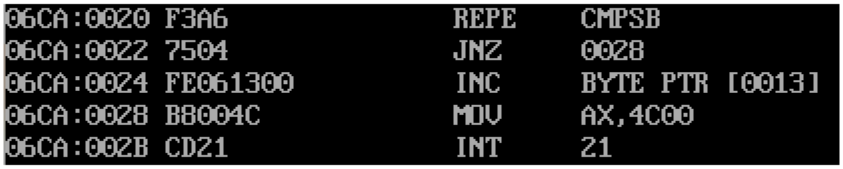
```
.8086
.model small
```

```
.data
    str1 db "naman"
    res db 0
    str2 db ?

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

    lea si, str1
    lea di, str2+4
    mov cx, 0005h
    up:      cld
           lodsb
           std
           stosb
           loop up

    cld
    lea si, str1
    lea di, str2
    mov cx, 0005h
    repe cmpsb
    jnz next
    inc res

    next: mov ax, 4c00h
    int 21h
    end
```

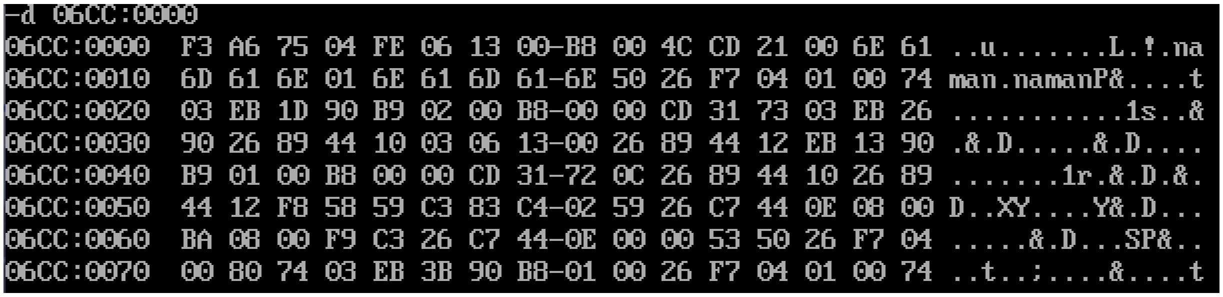
Output:



```
-u
06CA:0000 B8CC06            MOV     AX,06CC
06CA:0003 8ED8              MOV     DS,AX
06CA:0005 8EC0              MOV     ES,AX
06CA:0007 BE0E00            MOV     SI,000E
06CA:000A BF1800            MOV     DI,0018
06CA:000D B90500            MOV     CX,0005
06CA:0010 FC                CLD
06CA:0011 AC                LODSB
06CA:0012 FD                STD
06CA:0013 AA                STOSB
06CA:0014 E2FA              LOOPW   0010
06CA:0016 FC                CLD
06CA:0017 BE0E00            MOV     SI,000E
06CA:001A BF1400            MOV     DI,0014
06CA:001D B90400            MOV     CX,0004
```



```
06CA:0020 F3A6              REPE    CMPSB
06CA:0022 7504              JNZ     0028
06CA:0024 FE061300          INC     BYTE PTR [0013]
06CA:0028 B8004C            MOV     AX,4C00
06CA:002B CD21              INT     21
```



```
-d 06CC:0000
06CC:0000  F3 A6 75 04 FE 06 13 00-B8 00 4C CD 21 00 6E 61 ..u.......L.!.na
06CC:0010  6D 61 6E 01 6E 61 6D 61-6E 50 26 F7 04 01 00 74 man.namanP&....t
06CC:0020  03 EB 1D 90 B9 02 00 B8-00 00 CD 31 73 03 EB 26 ...........1s..&
06CC:0030  90 26 89 44 10 03 06 13-00 26 89 44 12 EB 13 90 .&.D.....&.D....
06CC:0040  B9 01 00 B8 00 00 CD 31-72 0C 26 89 44 10 26 89 .......1r.&.D.&.
06CC:0050  44 12 F8 58 59 C3 83 C4-02 59 26 C7 44 0E 08 00 D..XY....Y&.D...
06CC:0060  BA 08 00 F9 C3 26 C7 44-0E 00 00 53 50 26 F7 04 .....&.D...SP&..
06CC:0070  00 80 74 03 EB 3B 90 B8-01 00 26 F7 04 01 00 74 ..t..;....&....t
```

3. Write an assembly language program in 8086,

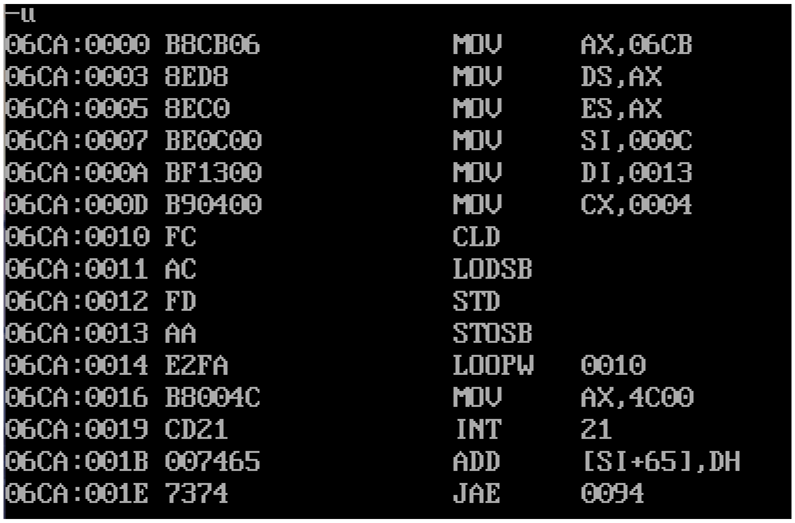
- To find reverse of an array

Code:

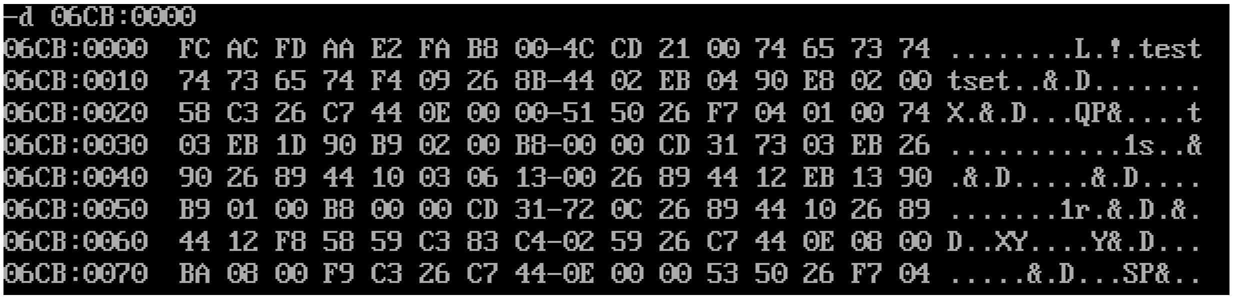
```
.8086
.model small

.data
   str1 db "test"
   str2 db ?

.code
```

```
mov ax, @data
mov ds, ax
mov es, ax

lea si, str1
lea di, str2+3
mov cx, 0004h
up:    cld
       lodsb
       std
       stosb
       loop up

mov ax, 4c00h
int 21h
end
```

Output:



```
-u
06CA:0000 B8CB06            MOV     AX,06CB
06CA:0003 8ED8              MOV     DS,AX
06CA:0005 8EC0              MOV     ES,AX
06CA:0007 BE0C00            MOV     SI,000C
06CA:000A BF1300            MOV     DI,0013
06CA:000D B90400            MOV     CX,0004
06CA:0010 FC                CLD
06CA:0011 AC                LODSB
06CA:0012 FD                STD
06CA:0013 AA                STOSB
06CA:0014 E2FA              LOOPW   0010
06CA:0016 B8004C            MOV     AX,4C00
06CA:0019 CD21              INT     21
06CA:001B 007465            ADD     [SI+65],DH
06CA:001E 7374              JAE     0094
```



```
-d 06CB:0000
06CB:0000  FC AC FD AA E2 FA B8 00-4C CD 21 00 74 65 73 74 ........L.!.test
06CB:0010  74 73 65 74 F4 09 26 8B-44 02 EB 04 90 E8 02 00 tset..&.D.......
06CB:0020  58 C3 26 C7 44 0E 00 00-51 50 26 F7 04 01 00 74 X.&.D...QP&....t
06CB:0030  03 EB 1D 90 B9 02 00 B8-00 00 CD 31 73 03 EB 26 ...........1s..&
06CB:0040  90 26 89 44 10 03 06 13-00 26 89 44 12 EB 13 90 .&.D.....&.D....
06CB:0050  B9 01 00 B8 00 00 CD 31-72 0C 26 89 44 10 26 89 .......1r.&.D.&.
06CB:0060  44 12 F8 58 59 C3 83 C4-02 59 26 C7 44 0E 08 00 D..XY....Y&.D...
06CB:0070  BA 08 00 F9 C3 26 C7 44-0E 00 00 53 50 26 F7 04 .....&.D...SP&..
```