PROGRAMS

1. Write an ALP to Display Fibonacci series

```
ASSUME CS:CODE, DS:DATA

DATA SEGMENT
COUNT DB 0AH
FIB DB 10 DUP(?)
DATA ENDS

CODE SEGMENT
START:
MOV AX, DATA
MOV DS, AX
MOV CL,COUNT

LEA SI,LIST

MOV AL,0H
MOV [SI],AL
MOV BL,01H

GO:   INC SI
      ADD BL,AL
      MOV [SI],BL
      MOV AL,[SI-1]
      DEC CL
      JNZ GO

INT 03H
CODE ENDS
END START
```

```
-U
0745:0000 B84407          MOV     AX,0744
0745:0003 8ED8            MOV     DS,AX
0745:0005 8A0E0000        MOV     CL,[0000]
0745:0009 BE0100          MOV     SI,0001
0745:000C B000            MOV     AL,00
0745:000E 8804            MOV     [SI],AL
0745:0010 B301            MOV     BL,01
0745:0012 46              INC     SI
0745:0013 02D8            ADD     BL,AL
0745:0015 881C            MOV     [SI],BL
0745:0017 8A44FF          MOV     AL,[SI-01]
0745:001A FEC9            DEC     CL
0745:001C 75F4            JNZ     0012
0745:001E CC              INT     3
```

Result

Data Segment

```
-D DS:0
0744:0000  0A 00 01 01 02 03 05 08-0D 15 22 37
```

2. Write an ALP to move a string of data bytes form one location to another

```
ASSUME CS:CODE, DS:DATA

DATA SEGMENT
STR1 DB 'CSE'
DATA ENDS

EXTRA SEGMENT
STR2 DB '00H'
EXTRA ENDS

CODE SEGMENT
START:

MOV AX, DATA
MOV DS, AX

MOV AX, EXTRA
MOV ES, AX

LEA SI,  STR1
LEA DI, STR2
MOV CL,03H

CLD
REP MOVSB

INT 03H
CODE ENDS
END START
```

```
-u
0746:0000 B84407          MOV     AX,0744
0746:0003 8ED8            MOV     DS,AX
0746:0005 B84507          MOV     AX,0745
0746:0008 8EC0            MOV     ES,AX
0746:000A BE0000          MOV     SI,0000
0746:000D BF0000          MOV     DI,0000
0746:0010 B103            MOV     CL,03
0746:0012 FC              CLD
0746:0013 F3A4            REP     MOVSB
0746:0015 CC              INT     3
```

Result

Data Segment

```
-d ds:0
0744:0000  43 53 45 00 00 00 00 00-00 00 00 00 00 00 00 00 CSE.....
```

Extra Segment

```
-d es:0
0745:0000  43 53 45 00 00 00 00 00-00 00 00 00 00 00 00 00 CSE.....
```

3. Write an ALP to concatenate two strings

```
ASSUME CS:CODE, DS:DATA

DATA SEGMENT
STR1 DB 'CSE'
STR2 DB 'SNIST'
STR3 DB 00H
DATA ENDS

CODE SEGMENT
START:
MOV AX,DATA
MOV DS,AX
MOV AX,0H

MOV CL,08H

LEA SI,STR1
LEA DI,STR3

GO:
MOV AL,[SI]
MOV [DI],AL
INC SI
INC DI
DEC CL
JNZ GO

INT 03H
CODE ENDS
END START
```

```
-u
0745:0000 B84407          MOV     AX,0744
0745:0003 8ED8            MOV     DS,AX
0745:0005 B80000          MOV     AX,0000
0745:0008 B108            MOV     CL,08
0745:000A BE0000          MOV     SI,0000
0745:000D BF0800          MOV     DI,0008
0745:0010 8A04            MOV     AL,[SI]
0745:0012 8805            MOV     [DI],AL
0745:0014 46              INC     SI
0745:0015 47              INC     DI
0745:0016 FEC9            DEC     CL
0745:0018 75F6            JNZ     0010
0745:001A CC              INT     3
```

Result

Data Segment

```
-d ds:0
0744:0000  63 73 65 73 6E 69 73 74-63 73 65 73 6E 69 73 74 csesnistcsesnist
```

4. Write an ALP to reverse a given string

```
ASSUME CS:CODE, DS:DATA,

DATA SEGMENT
STR1 DB 'CSE'
DATA ENDS


CODE SEGMENT
START:
MOV AX, DATA
MOV DS, AX
MOV AX,0H

MOV CL,03H

LEA SI,STR1
LEA DI,STR+6

GO:
MOV AL,[SI]
MOV [DI],AL
INC SI
DEC DI
DEC CL
JNZ GO

INT 03H
CODE ENDS
END START
```

```
-u
0745:0000 B84407          MOV     AX,0744
0745:0003 8ED8            MOV     DS,AX
0745:0005 B80000          MOV     AX,0000
0745:0008 B103            MOV     CL,03
0745:000A BE0000          MOV     SI,0000
0745:000D BF0600          MOV     DI,0006
0745:0010 8A04            MOV     AL,[SI]
0745:0012 8805            MOV     [DI],AL
0745:0014 46              INC     SI
0745:0015 4F              DEC     DI
0745:0016 FEC9            DEC     CL
0745:0018 75F6            JNZ     0010
0745:001A CC              INT     3
```

Result

Data Segment

```
-d ds:0
0744:0000  43 53 45 00 45 53 43 00-00 00 00 00 00 00 00 00 CSE.ESC.
```