## **Palindrome**

palindrome.asm

```
DATA SEGMENT
   String1 DB "MALAYALAM", "$"
   PALINMSG DB "Palindrome$"
   NONPALINMSG DB "Not Palindrome$"
DATA ENDS
EXTRA SEGMENT
   String2 DB 09H DUP (?)
EXTRA ENDS
CODE SEGMENT
   ASSUME CS:CODE, DS:DATA, ES:EXTRA
PRINTMSG MACRO MSG
   MOV DX, OFFSET MSG
   MOV AH, 09H
   INT 21H
ENDM
START:
   MOV AX, DATA
   MOV DS, AX
   MOV AX, EXTRA
   MOV ES, AX
   LEA SI, String1
   LEA DI, String2 + 08H
   MOV CX, 0009H
REVLOOP:
   CLD
   LODSB
   STD
    STOSB
```

```
LOOP REVLOOP
   LEA SI, String1
   LEA DI, String2
   MOV CX, 0009H
   CLD
   REPE CMPSB
   JNZ SKIP
   PRINTMSG PALINMSG
   JMP FIN
SKIP:
   PRINTMSG NONPALINMSG
FIN:
   MOV AH, 4CH
   INT 21H
CODE ENDS
END START
```

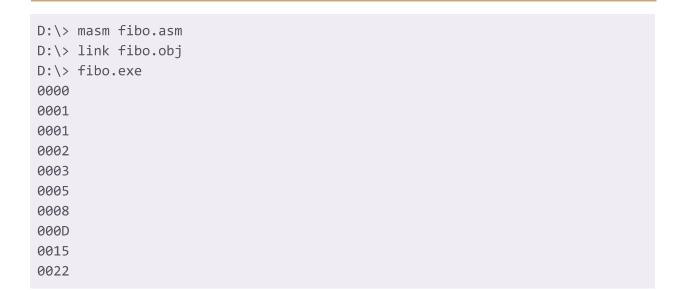
```
D:\> masm Palindrome.asm
D:\> link palindrome.obj
D:\> palindrome.exe
PALINDROME
```

## **Fibonacci**

fibo.asm

```
DATA SEGMENT
   COUNT DB OAH
   SEP DB OAH, "$"
   A DW 0000H
   B DW 0001H
   TEMP DW ?
DATA ENDS
CODE SEGMENT
   ASSUME CS:CODE, DS:DATA
PRINTMSG MACRO MSG
   LEA DX, MSG
   MOV AH, 09H
   INT 21H
ENDM
DECHEX PROC
   MOV DX, 000FH
   AND DX, AX
   MOV CL, 04
   ROR AX,CL
   CMP DL,09H
   JLE HEXSKIP
   ADD DL, 07H
HEXSKIP:
   ADD DL,30H
   RET
DECHEX ENDP
PRINTNUM PROC
   MOV CL,12
   ROR AX,CL
   MOV SI,AX
   MOV BL,04H
JBCK:
```

```
CALL DECHEX
   MOV AH, 02H
   INT 21H
   MOV AX,SI
   MOV CL, 04
   ROL AX,CL
   MOV SI, AX
   DEC BL
   JNZ JBCK
   RET
PRINTNUM ENDP
START:
   MOV AX, DATA
   MOV DS, AX
INIT:
   MOV CL, COUNT
   CMP CL, 00H
   JLE LAST
   DEC CL
   MOV COUNT, CL
   MOV AX, A
   ADD AX, B
   MOV TEMP, AX
   MOV AX,A
   CALL PRINTNUM
   PRINTMSG SEP
   MOV AX, B
   MOV A, AX
   MOV AX, TEMP
   MOV B, AX
   JMP INIT
LAST:
   MOV AH, 4CH
   INT 21H
CODE ENDS
END START
```



## Sum of Two 16 bit Numbers

add.asm

```
DATA SEGMENT
   A DW 6667H
   B DW 9999H
   SUM DW 0000H
   CARRY DB 00H
   MSG1 DB "SUM OF $"
    MSG2 DB " AND $"
   MSG3 DB " = $"
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
PRINTMSG MACRO MSG
   LEA DX, MSG
   MOV AH, 09H
   INT 21H
ENDM
DECHEX PROC
   MOV DX, 000FH
   AND DX, AX
   MOV CL, 04
   ROR AX, CL
   CMP DL,09H
    JLE HEXSKIP
   ADD DL, 07H
HEXSKIP:
   ADD DL,30H
   RET
DECHEX ENDP
PRINTNUM PROC
   MOV CL,12
    ROR AX, CL
    MOV SI, AX
```

```
MOV BL,04H
JBCK:
   CALL DECHEX
   MOV AH, 02H
   INT 21H
   MOV AX,SI
   MOV CL, 04
   ROL AX,CL
   MOV SI, AX
   DEC BL
    JNZ JBCK
   RET
PRINTNUM ENDP
START:
   MOV AX, DATA
   MOV DS, AX
   MOV AX, B
   ADD AX,A
    JNC SKIPCARRY
   MOV CARRY, 1H;
SKIPCARRY:
   MOV SUM, AX
   PRINTMSG MSG1
   MOV AX, A
   CALL PRINTNUM
   PRINTMSG MSG2
   MOV AX, B
   CALL PRINTNUM
   PRINTMSG MSG3
   MOV AL, CARRY
   CMP AL,00H
   JE NOCARRY
   MOV DL,31H
   MOV AH,02H
   INT 21H
NOCARRY:
   MOV AX, SUM
   CALL PRINTNUM
   MOV AH, 4CH
   INT 21H
```

CODE ENDS END START

D:\> masm add.asm

D:\> link add.obj

D:\> add.exe

SUM OF 6667 AND 9999 = 10000