

Microprocessor LAB Codes

ASIF ZAMAN RIZVE

- 1. Two Single Digit Number Addition
- 2. Two Single Digit Number Subtraction
- 3. Two Single Digit Number Division
- 4. Two Single Digit Number Multiplication
- 5. Odd Even check of a Number
- 6. Largest Number among The Numbers
- 7. Prime Non-Prime Number Check
- 8. Summation of 1-9 numbers
- 9. Display a String
- 10. String Lower to Upper or Upper to Lower Case
- 11. Display String Vertically
- 12. String Reverse Display
- 13. Multiple Same letter Finding in a String
- 14. Mismatch of two String
- 15. Put Large Number on CL Register
- 16. Addition of 25,12,15,10,11
- 17. BCD to BINARY
- 18. BINARY to BCD
- $19. 1^2 + 3^2 + 5^2 + 7^2$

1. Two Single Digit Number Addition:

Solution:

First Add both numbers then use DAA instruction to find the sum if the addition result exceeds 9. after DAA the result will be stored in AL register.

```
.MODEL SMALL
                                                GREATER:
.STACK 100
                                                    SUB AL, 10H
.DATA
                                                    MOV BL, AL
.CODE
                                                   MOV AH,02H
MAIN PROC
                                                   MOV DL,01H
    MOV AH,01H
                                                    ADD DL,30H
    INT 21H
                                                    INT 21H
                                                    MOV AH,02H
    SUB AL, 30H
    MOV BL,AL
                                                    MOV DL,BL
                                                   ADD DL,30H
INT 21H
    MOV AH,01H
    INT 21H
                                                   JMP EXIT
    SUB AL, 30H
                                               EQUAL:
                                                    MOV AH,02H
                                                   MOV DL,01H
    ADD AL, BL
                                                   ADD DL, 30H
    MOV CL, AL
                                                    INT 21H
                                                    MOV AH,02H
                                                   MOV DL,00H
    MOV AH,02H
    MOV DL, OAH
                                                    ADD DL,30H
    INT 21H
                                                    INT 21H
    MOV AH,02H
                                                    JMP EXIT
    MOV DL, ODH
                                                    NOT_GREATER:
    INT 21H
    MOV AL, CL
                                                    MOV AH, 02H
                                                   MOV DL,AL
    CMP AL, 10H
                                                    ADD DL, 30H
    JE EQUAL
    JNG NOT GREATER
                                                    INT 21H
                                               EXIT:
                                               MAIN ENDP
                                               END MAIN
```

2. Two Single Digit Number Subtraction:

Solution:

First Subtract two numbers. if the answer is negative the sign flag will be set. then sign jump will be done and 2's complement will be taken by using NEG instruction. if the result of subtraction is positive then nothing will be happened.

if the result of subtraction is positive then	i nouning with be nappened.
.MODEL SMALL	NEGATIVE:
.STACK 100	MOV DX,13;
.DATA	MOV AH, 2
.CODE	INT 21H
MAIN PROC	MOV DX,10;
	MOV AH, 2
MOV AH,01H	INT 21H
INT 21H	NEG BL
SUB AL,30H	MOV DL,2DH
MOV BL, AL	MOV AH,02H
MOV CL, BL	INT 21H
	MOV DL,BL
MOV AH,01H	ADD DL,30H
INT 21H	MOV AH,02H
SUB AL,30H	INT 21H
	JMP EXIT
SUB BL, AL	
JS NEGATIVE	EXIT:
MOV DL, BL	MAIN ENDP
ADD DL, 30H	END MAIN
MOV BL, DL	
MOV DX,13;	
MOV AH, 2	
INT 21H	
MOV DX,10;	
MOV AH, 2	
INT 21H	
MOV DL, BL	
MOV AH,02H INT 21H	
JMP EXIT	
OHI BAII	

3. Two Single Digit Number Division: Solution:

.MODEL SMALL	MOV AH,02H
.STACK 100	MOV DL,13
.CODE	INT 21H
MAIN PROC	MOV AH,02H
MOV AH,01H	MOV DL,10
INT 21H	INT 21H
SUB AL, 30H	
MOV CL, AL	MOV AH,02H
	MOV DL,CL
MOV AH,02H	ADD DL,30H
MOV DL,13	INT 21H
INT 21H	
MOV AH,02H	MOV AH,02H
MOV DL, 10	MOV DL,13
INT 21H	INT 21H
	MOV AH,02H
MOV AH,01H	MOV DL,10
INT 21H	INT 21H
SUB AL, 30H	
MOV BL, AL	MOV AH,02H
MOV AL, CL	MOV DL, CH
MOV AH,00H	ADD DL,30H
DIV BL; AL/BL Result: AL Remainder: AH	INT 21H
MOV CL, AL	
MOV CH, AH	MAIN ENDP
	END MAIN

4. Two Single Digit Number Multiplication: Solution:

```
.MODEL SMALL
                                                      MOV BL, AL
.STACK 100
                                                     MOV BH, AH
                                                     MOV AH,02H
.DATA
.CODE
                                                     MOV DL, OAH
                                                     INT 21H
MAIN PROC
   MOV AH,01H
                                                     MOV AH,02H
                                                     MOV DL, ODH
INT 21H
    INT 21H
    SUB AL, 30H
    MOV CL, AL
                                                     MOV AH, 02H
    MOV AH,02H
                                                     ADD BH,30H
    MOV DL,13
                                                     MOV DL, BH
                                                     INT 21H
    INT 21H
    MOV AH,02H
                                                     MOV AH,02H
                                                     ADD BL,30H
MOV DL,BL
    MOV DL,10
    INT 21H
    MOV AH,01H
                                                     INT 21H
    INT 21H
                                                 MAIN ENDP
    SUB AL, 30H
                                                 END MAIN
    MOV BL,AL
    MOV AL, CL
    MUL BL
    AAM
```

5. Odd Even Check: Solution:

```
.MODEL SMALL
                                                   INT 21H
.STACK 100
                                                   MOV AX, @DATA
.DATA
                                                   MOV DS, AX
MSG1 DB "ENTER THE NUMBER:$"
                                                   LEA DX, MSG3
MSG2 DB "THE NUMBER IS EVEN$"
                                                   MOV AH,09H
MSG3 DB "THE NUMBER IS ODD$"
                                                   INT 21H
.CODE
                                                   JMP EXIT
MAIN PROC
                                                   EVEN:
   MOV AX, @DATA
                                                   MOV AH,02H
    MOV DS, AX
                                                   MOV DL, OAH
    LEA DX, MSG1
                                                   INT 21H
    MOV AH,09H
                                                   MOV AH,02H
    INT 21H
                                                   MOV DL, ODH
    MOV AH,01H
                                                   INT 21H
    INT 21H
                                                   MOV AX, @DATA
    SUB AL, 30H
                                                   MOV DS, AX
    MOV BL,02H
                                                   LEA DX, MSG2
    DIV BL
                                                   MOV AH, 09H
    CMP AH,00H
                                                   INT 21H
    JE EVEN
                                                   JMP EXIT
    ODD:
                                               EXIT:
    MOV AH,02H
                                               MAIN ENDP
    MOV DL, OAH
                                               END MAIN
    INT 21H
    MOV AH,02H
    MOV DL, ODH
```

6. Largest Number among The Numbers: Solution:

```
.MODEL SMALL
.STACK 100
.DATA
LIST DB 25H, 12H, 35H, 10H, 11H, 50H, 48H, 60H
.CODE
MAIN PROC
   MOV AX, @DATA
  MOV DS, AX
  LEA SI,LIST
   MOV CL,7
  MOV AL, [SI]
   TOP:
   MOV BL,[SI]
   CMP AL, BL
   JB NEXT:
   INC SI
   LOOP TOP
   JMP EXIT
   NEXT:
   MOV AL, BL
   INC SI
   JMP TOP
EXIT:
MAIN ENDP
END MAIN
```

7. Prime Non-Prime Check: Solution:

```
NOT_PRIME:
MOV AH,02H
.MODEL SMALL
.STACK 100
                                                   MOV DL, OAH
.DATA
MSG0 DB "ENTER THE NUMBER:$"
                                                   INT 21H
                                                   MOV AH, 02H
MSG1 DB "PRIME$"
MSG2 DB "NOT_PRIME$"
                                                   MOV DL, ODH
                                                   INT 21H
.CODE
MAIN PROC
                                                   MOV AX, @DATA
    MOV AX,@DATA
                                                   MOV DS, AX
    MOV DS, AX
                                                   LEA DX, MSG2
    LEA DX, MSG0
                                                   MOV AH,09H
    MOV AH,09H
                                                   INT 21H
    INT 21H
                                                   JMP EXIT
    MOV AH,01H
    INT 21H
                                                   PRIME:
    SUB AL,30H
                                                   MOV AH,02H
                                                   MOV DL, OAH
    CMP AL,02H
                                                   INT 21H
                                                   MOV AH, 02H
    JE PRIME
    CMP AL,01H
                                                   MOV DL, 0DH
    JE EXIT
                                                   INT 21H
    CMP AL, 00H
                                                   MOV AX, @DATA
                                                   MOV DS, AX
    JE EXIT
                                                   LEA DX, MSG1
    MOV CL,AL
                                                   MOV AH,09H
                                                   INT 21H
    MOV CH, AL
    SUB CH,01H
                                                   JMP EXIT
    MOV BL,2
                                               EXIT:
                                               MOV AH, 4CH
    MOV AH,00H
                                               INT 21H
    MOV AL, CL
                                               MAIN ENDP
    DIV BL
                                               END MAIN
    CMP AH,00H
    JE NOT_PRIME
    CMP BL, CH
    JE PRIME
    INC BL
    JMP TOP
```

8. Summation of 1-9 numbers: Solution:

```
.MODEL SMALL
.STACK 100
.CODE

MAIN PROC

MOV BL,01H ;Start of Adding Value

MOV AL,00H ;Initial Digit

MOV CL,09H ;Counter

TOP:

ADD AL,BL ;Add

DAA ;Decimal Addition Adjust
INC BL ;Increment

LOOP TOP

MAIN ENDP
END MAIN
```

9. Display a String: Solution:

```
.MODEL SMALL
.STACK 100
.DATA
STRING1 DB "MARS$"
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
LEA DX,STRING1
MOV AH,09H
INT 21H
EXIT:
MAIN ENDP
END MAIN
```

10. String Upper to Lower and Lower to Upper: Solution:

```
.MODEL SMALL
                                                      DO CAPITAL:
.STACK 100
                                                      \overline{\text{MOV}} BL, [SI]
.DATA
                                                      SUB BL,20H
STRING1 DB "hElLo$"
                                                      MOV [SI],BL
                                                      JMP NEXT:
.CODE
MAIN PROC
                                                      DO SMALL:
    MOV AX,@DATA
                                                      MOV BL, [SI]
    MOV DS, AX
                                                      ADD BL, 20H
                                                      MOV [SI],BL
    LEA SI, STRING1
                                                      NEXT:
    TOP:
                                                      INC SI
    CMP [SI],'$'
                                                      JMP TOP
    JZ EXIT
                                                  EXIT:
    CMP [SI],41H
    JL EXIT
                                                  LEA DX, STRING1
    CMP [SI],5AH
                                                  MOV AH,09
    JLE DO SMALL:
                                                  INT 21H
    CMP [S\overline{I}], 61H
                                                  MAIN ENDP
                                                  END MAIN
    JL EXIT
    CMP [SI],7AH
    JG EXIT
```

11. Display String Vertically Solution:

```
.MODEL SMALL
.STACK 100
                                                  MOV AH,02H
                                                  MOV DL,13
.DATA
STRING1 DB "HELLO$"
                                                  INT 21H
                                                  MOV AH,02H
.CODE
                                                  MOV DL,10
MAIN PROC
   MOV AX, @DATA
                                                  INT 21H
   MOV DS, AX
                                                  INC SI
    LEA SI, STRING1
                                                  JMP TOP
    TOP:
                                              EXIT:
    CMP [SI],'$'
                                              MAIN ENDP
    JZ EXIT
                                              END MAIN
    MOV AH,02H
    MOV DL,[SI]
    INT 21H
```

12. String Reverse Display Solution:

12.1. Using PUSH POP instruction:

```
.MODEL SMALL
                                               INC SI
.STACK 100
                                                  LOOP TOP
.DATA
                                                  MOV CL,00H
STRING1 DB "HELLO"
                                                  MOV CL,05H
.CODE
                                                  LOOP1:
MAIN PROC
                                                  POP DX
   MOV AX, @DATA
                                                  MOV AH,02H
    MOV DS,AX
                                                   INT 21H
    LEA SI, STRING1
                                                  LOOP LOOP1
    MOV CL,05H
    TOP:
                                               EXIT:
    PUSH [SI]
                                              MAIN ENDP
                                           END MAIN
```

12.2. Using Direction Flag SET

```
TOP:
.MODEL SMALL
.STACK 100H
                                                    CMP DI,00H
                                                    JE EXIT
.DATA
STRING1 DB 'HELLO'
                                                    MOVSB
STRING2 DB 5 DUP(?)
                                                    MOV DL, [DI]
.CODE
                                                    MOV AH, 2
                                                    INT 21H
MAIN PROC
    MOV AX,00H
                                                    JMP TOP
    MOV AX,@DATA
                                                    EXIT:
    MOV DS, AX
                                                MAIN ENDP
    MOV ES, AX
                                            END MAIN
    LEA SI, STRING1
    LEA DI, STRING2
    STD
```

13. Multiple Same letter Finding in a String Solution:

```
.MODEL SMALL
                                                        COUNT:
                                                       INC CL
.STACK 100
.DATA
                                                       NEXT:
STRING1 DB 'ZAMAAAN$'
                                                       INC SI
.CODE
                                                   JMP TOP
MAIN PROC
                                               EXIT:
                                               MOV AH,02H
    MOV AX,@DATA
    MOV DS, AX
                                               MOV DL,13
                                               INT 21H
    LEA SI, STRING1
    MOV AH,01H
                                               MOV AH,02H
                                               MOV DL,10
    INT 21H
    MOV BL, AL
                                               INT 21H
    MOV CL,00H
                                               MOV AH,02H
                                               ADD CL, 30H
        CMP [SI],'$'
                                               MOV DL, CL
                                               INT 21H
        JZ EXIT
        CMP [SI], BL
                                               MAIN ENDP
        JNE NEXT
                                               END MAIN
```

14. Mismatch between two Strings Solution:

```
.MODEL SMALL
                                                   DONE:
.STACK 100
                                                   MOV AX, @DATA
                                                   MOV DS, AX
.DATA
STRING1 DB "HELLO$"
                                                   LEA DX, STRING3
STRING2 DB "HEPLO$"
                                                  MOV AH,09H
STRING3 DB "NO MISMATCH BETWEEN TWO
                                                  INT 21H
STRINGS$"
                                                   JMP EXIT
                                                  NOT DONE:
STRING4 DB "THERE IS MISMATCH BETWEEN TWO
STRINGS$"
                                                  MOV AX, @DATA
.CODE
                                                  MOV DS, AX
MAIN PROC
                                                   LEA DX, STRING4
                                                  MOV AH, 09H
   MOV AX, @DATA
   MOV DS, AX
                                                   INT 21H
   MOV ES, AX
                                                   JMP EXIT
    LEA SI, STRING1
                                               EXIT:
   LEA DI, STRING2
                                               MAIN ENDP
                                               END MAIN
    TOP:
    CMP [SI],'$'
    JZ DONE
    LODSB
    CMP [DI], AL
    JNE NOT_DONE
    INC DI
    JMP TOP
```

15. Put Large Number on CL Register Solution:

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
    MOV AH,01H
    INT 21H
    SUB AL,30H
    MOV BL, AL
    MOV AH,01H
    INT 21H
    SUB AL, 30H
    CMP BL, AL
    JE EQUAL
    JG LARGE BL
    LARGE AL:
    MOV CL, AL
    JMP EXIT
    LARGE BL:
    MOV CL, BL
    JMP EXIT
    EQUAL:
    MOV CL,AL
    EXIT:
    MAIN ENDP
END MAIN
```

16. Addition of 25,12,15,10,11 Solution:

```
.MODEL SMALL
.STACK 100
.DATA
LIST DB 25H, 12H, 15H, 10H, 11H
.CODE
MAIN PROC
   MOV AX,@DATA
   MOV DS, AX
   LEA DX, LIST
   MOV CL,5
    MOV AL,00H
    TOP:
    ADD AL, [SI]
    DAA
    INC SI
   LOOP TOP
MAIN ENDP
END MAIN
```

17. BCD to Binary Conversion Solution:

```
.MODEL SMALL
.STACK 100H
                                               MOV AX, DATA1
.DATA
                                               AND AX,00FFH
BCD DW 0456H
                                               AND AL, OFH
                                               MOV CH,AL
BIN DW ?
.CODE
                                               MOV AX,00H
                                               MOV BL,01H
MAIN PROC
DATA1 EQU 0456H
                                               MOV AL, CH
                                               MUL BL
MOV AX, DATA1
AND AX, OFFOOH
                                               ADD DX, AX
MOV CH, AH
MOV AX,00H
                                               MOV AX,@DATA
                                               MOV DS, AX
MOV DX,00H
MOV BL,64H
                                               MOV BIN, DX
MOV AL, CH
                                               MAIN ENDP
MUL BL
                                               END MAIN
MOV DX, AX
MOV AX, DATA1
AND AX,00FFH
AND AL, OFOH
ROL AL,04H
MOV CH, AL
MOV AX,00H
MOV BL, OAH
MOV AL, CH
MUL BL
ADD DX, AX
```

$19. 1^2 + 3^2 + 5^2 + 7^2$

Solution:

.MODEL SMALL	MOV DL, AL
.STACK 100H	MUL AL
.DATA	ADD BX, AX
.CODE	MOV AL, DL
MAIN PROC	ADD AL,02H
MOV AL,01H	LOOP TOP
MOV CX,05H	MOV AX, BX
TOP:	DAA
	MAIN ENDP
	END MAIN