**Microprocessor Practical**

# Q1 Program to:

1. **ADD Two 32 Bit Binary Numbers.**

.MODEL SMALL

.386

.DATA

MES1 DB 13,10,'ENTER FIRST NUMBER : $' MES2 DB 13,10,'ENTER SECOND NUMBER : $' MES3 DB 13,10,'32 ADDITION IS : $'

.CODE

.STARTUP

MOV AH,09H

MOV DX, OFFSET MES1 INT 21H

MOV CX,8

I1: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I1

MOV ESI,EBX

MOV EBX,00000000H

MOV AH,09H

MOV DX, OFFSET MES2 INT 21H

MOV CX,8

I2: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I2

MOV AH,09H

MOV DX, OFFSET MES3

INT 21H

MOV EAX,ESI ADD EAX,EBX CALL DISPH

.EXIT

DISPH PROC NEAR PUSH EBX

PUSH ECX MOV CL,4 MOV CH,8

DISPH1:

ROL EAX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP ECX POP EBX RET

DISPH ENDP END

# SUB two 32 bit binary number

.MODEL SMALL

.386

.DATA

MES1 DB 13,10,'ENTER FIRST NUMBER : $' MES2 DB 13,10,'ENTER SECOND NUMBER : $' MES3 DB 13,10,'32 BIT BCD SUBTRACTION IS : $'

.CODE

.STARTUP

MOV AH,09H

MOV DX, OFFSET MES1 INT 21H

MOV CX,8

I1: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I1

MOV ESI,EBX MOV EBX,0

MOV AH,09H

MOV DX, OFFSET MES2 INT 21H

MOV CX,8

I2: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I2

MOV AH,09H

MOV DX, OFFSET MES3 INT 21H

MOV EAX,ESI SUB EAX,EBX CALL DISPH

.EXIT

DISPH PROC NEAR PUSH EBX

PUSH ECX MOV CL,4 MOV CH,8

DISPH1:

ROL EAX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP ECX POP EBX RET

DISPH ENDP END

# MUL two 32 bit Binary Numbers

.MODEL SMALL

.386

.DATA

MES1 DB 13,10,'ENTER FIRST NUMBER : $' MES2 DB 13,10,'ENTER SECOND NUMBER : $' MES3 DB 13,10,'PRODUCT IS : $'

.CODE

.STARTUP

MOV AH,09H

MOV DX, OFFSET MES1 INT 21H

MOV CX,8

I1: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I1

MOV ESI,EBX MOV EBX,0

MOV AH,09H

MOV DX, OFFSET MES2 INT 21H

MOV CX,8

I2: SHL EBX,4 MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP I2

MOV AH,09H

MOV DX, OFFSET MES3 INT 21H

MOV EAX,ESI MUL EBX MOV ECX,EAX MOV EAX,EDX CALL DISPH MOV EAX,ECX CALL DISPH

.EXIT

DISPH PROC NEAR

PUSH EBX PUSH ECX MOV CL,4 MOV CH,8

DISPH1:

ROL EAX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP ECX POP EBX RET

DISPH ENDP END

# DIV of two 32 bit Binary Numbers

.MODEL SMALL

.386P

.DATA ;

N1 DD 1234567AH N2 DD 11111111H N3 DD ?

N4 DD ? COUNT DB 08H

QUOTIENT DB "QUOTIENT:", 10, 13, '$'

REMAINDER DB "REMAINDER:", 10, 13, '$'

.CODE ;

.STARTUP ;

MOV AX, @DATA MOV DS, AX

MOV EDX, 00000000H MOV EAX, N1

DIV N2

MOV N3, EDX MOV N4, EAX

MOV DX, OFFSET QUOTIENT MOV AH, 09H

INT 21H

MOV EAX, N4 CALL DISP MOV DL, 13 MOV AH, 02H INT 21H

MOV DL, 10 MOV AH, 02H INT 21H

MOV DX, OFFSET REMAINDER MOV AH, 09H

INT 21H

MOV EAX, N3 CALL DISP MOV AH, 4CH INT 21H

DISP PROC NEAR MOV CL, COUNT UP:

ROL EAX, 04H MOV BX, AX AND AL, 0FH CMP AL, 0AH

JNGE NOTGREATER

ADD AL, 07H NOTGREATER: ADD AL, 30H

MOV DL, AL MOV AH, 02H INT 21H

RET

MOV AX, BX LOOP UP

END

.EXIT

ENDP DISP

# Q2 32 Bit

1. **BCD Addition**

.MODEL SMALL

.386

.DATA

M1 DB 10,13,'ENTER THE FIRST NO. $' M2 DB 10,13,'ENTER THE SECOND NO. $' M3 DB 10,13,'BCD ADDITION IS $'

.CODE

.STARTUP MOV ESI,0 MOV EDX,0 MOV EBX,0 MOV EAX,0 MOV EBP,0 MOV ECX,0

MOV AH,9

MOV DX,OFFSET M1 INT 21H

MOV CX,8 L1:SHL EBX,4

MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP L1

MOV ESI,EBX MOV EBX,0

MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV CX,8 L3:SHL EBX,4

MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP L3

MOV EDX,EBX MOV EBX,ESI

MOV AL,BL ADD AL,DL DAA

MOV CL,AL

MOV AL,BH ADC AL,DH DAA

MOV CH,AL

LAHF

ROL EBX,16 ROL EDX,16 ROL ECX,16 SAHF

MOV AL,BL ADC AL,DL DAA

MOV CL,AL

MOV AL,BH ADC AL,DH DAA

MOV CH,AL

ROL ECX,16 MOV EBX,0 MOV EBX,ECX

MOV AH,9

MOV DX,OFFSET M3 INT 21H

MOV CX,8

L5: ROL EBX,4 MOV DL,BL AND DL,0FH ADD DL,30H MOV AH,2 INT 21H LOOP L5

.EXIT END

# BCD Subtraction

.MODEL SMALL

.386

.DATA

M1 DB 10,13,'ENTER THE FIRST NO. $' M2 DB 10,13,'ENTER THE SECOND NO. $' M3 DB 10,13,'BCD SUBTRACTION IS $'

.CODE

.STARTUP MOV ESI,0 MOV EDX,0 MOV EBX,0 MOV EAX,0 MOV EBP,0 MOV ECX,0

MOV AH,9

MOV DX,OFFSET M1 INT 21H

MOV CX,8 L1:SHL EBX,4

MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP L1

MOV ESI,EBX MOV EBX,0

MOV AH,9

MOV DX,OFFSET M2

INT 21H

MOV CX,8 L3:SHL EBX,4

MOV AH,1 INT 21H SUB AL,30H ADD BL,AL LOOP L3

MOV EDX,EBX MOV EBX,ESI MOV AL,BL SUB AL,DL DAS

MOV CL,AL MOV AL,BH SBB AL,DH DAS

MOV CH,AL LAHF

ROL EBX,16 ROL EDX,16 ROL ECX,16 SAHF

MOV AL,BL SBB AL,DL DAS

MOV CL,AL MOV AL,BH SBB AL,DH DAS

MOV CH,AL ROL ECX,16 MOV EBX,0 MOV EBX,ECX

MOV AH,9

MOV DX,OFFSET M3 INT 21H

MOV CX,8

L5: ROL EBX,4 MOV DL,BL AND DL,0FH ADD DL,30H

MOV AH,2 INT 21H LOOP L5

.EXIT END

# Q3 Linear Search

.MODEL SMALL

.386

.DATA

ARRAY DB 50 DUP(?)

M1 DB 10,13,'ENTER THE SIZE OF ARRAY$' M2 DB 10,13,'ENTER DIGIT$'

M3 DB 10,13,'ENTER THE KEY TO BE SEARCHED$' M4 DB 10,13,'KEY FOUND :)$'

M5 DB 10,13,'KEY NOT FOUND :($' M6 DB 10,13,'POSITION: $'

NUM DW 0H KEY DW 0H POS DW 0H

.CODE

.STARTUP

MOV AX,0H MOV CX,0H MOV DI,0H MOV BX,0H

MOV AH,9

MOV DX,OFFSET M1 INT 21H

MOV AH,1 INT 21H SUB AL,30H MOV AH,0

MOV NUM,AX MOV DI,0H MOV CX,NUM

L1:MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV ARRAY[DI],AL INC DI

LOOP L1

MOV AH,9

MOV DX,OFFSET M3 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV AH,0 MOV KEY,AX MOV CX,NUM MOV DI,0 MOV DX,KEY SEARCH:

MOV BL,ARRAY[DI] MOV BH,0

CMP BX,DX

JE FOUND

INC DI DEC CX CMP CX,0

JNZ SEARCH JMP NOTFOUND

FOUND:

MOV AH,9

MOV DX,OFFSET M4 INT 21H

MOV AX,BX CALL DISPH

MOV AH,9

MOV DX,OFFSET M6 INT 21H

MOV AX,CX CALL DISPH CALL q

NOTFOUND:MOV AH,9 MOV DX,OFFSET M5 INT 21H

q:

.EXIT

DISPH PROC NEAR

PUSH AX PUSH CX MOV CL,4 MOV CH,4

DISPH1:

ROL AX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP CX POP BX RET

DISPH ENDP END

# Q4 Binary Search

.MODEL SMALL

.DATA

D1 DB 13,10,'Enter the element:$'

D2 DB 13,10,'Enter the element to be searched:$' D3 DB 13,10,'Element not found.$'

D4 DB 13,10,'Element found.$'

D5 DB 13,10,'Enter the size of the array:$' ARRAY DB 20 DUP(?)

TEMP DB ? L DB ?

U DB ?

.CODE

.STARTUP

MOV DX,OFFSET D5 MOV AH,09H

INT 21H

MOV AH,01H INT 21H

SUB AL,30H MOV BL,AL

MOV TEMP,BL DEC BL

MOV U,BL

MOV CH,0 MOV CL,TEMP

MOV DX,OFFSET D1 MOV SI,0

L1:

MOV AH,09H INT 21H MOV AH,01H INT 21H

SUB AL,30H MOV BL,AL INT 21H SUB AL,30H SHL BL,4

ADD BL,AL

MOV ARRAY[SI],BL INC SI

LOOP L1

MOV DX,OFFSET D2 MOV AH,09H

INT 21H

MOV AH,01H INT 21H

SUB AL,30H

MOV BL,AL INT 21H SUB AL,30H SHL BL,4 ADD BL,AL

MOV CL,TEMP MOV L,0H

L2:

MOV AH,0H MOV AL,L ADD AL,U MOV BH,02H DIV BH MOV AH,0H MOV SI,AX

CMP ARRAY[SI],BL JA A

JB B

MOV DX,OFFSET D4 MOV AH,09H

INT 21H JMP E MOV AX,SI A:

MOV U,AL JMP L9

B:

MOV L,AL

L9:

MOV AL,L CMP AL,U

LOOPNE L2

MOV DX,OFFSET D3 MOV AH,09H

INT 21H JMP E

E:

.EXIT END

# Q5 Two Arrays

1. **Addition**

.MODEL SMALL

.386

.DATA

ARRAY1 DB 50 DUP(?) ARRAY2 DB 50 DUP(?) RESULT DB 50 DUP(?)

M0 DB 10,13,'PROGRAM TO ADD 2 ARRAYS$' M1 DB 10,13,'ENTER THE SIZE OF ARRAYS$' M2 DB 10,13,'ENTER DIGIT$'

M3 DB 10,13,'ENTER ELEMENT OF ARRAY1$' M4 DB 10,13,'ENTER ELEMENT OF ARRAY2$' M5 DB 10,13,'RESULTING ARRAY$'

NUM DW 0H

.CODE

.STARTUP

MOV AX,0H MOV CX,0H MOV DI,0H MOV BX,0H

MOV AH,9

MOV DX,OFFSET M0 INT 21H

MOV AH,9

MOV DX,OFFSET M1 INT 21H

MOV AH,1 INT 21H SUB AL,30H MOV AH,0

MOV NUM,AX MOV DI,0H MOV CX,NUM

MOV AH,9

MOV DX,OFFSET M3 INT 21H

L1:MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV ARRAY1[DI],AL INC DI

LOOP L1

MOV DI,0 MOV CX,NUM

MOV AH,9

MOV DX,OFFSET M4 INT 21H

L2:MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV ARRAY2[DI],AL INC DI

LOOP L2

MOV DI,0 MOV CX,NUM

L3:

MOV DL,ARRAY1[DI]

ADD DL,ARRAY2[DI] MOV RESULT[DI],DL INC DI

LOOP L3

MOV AH,9

MOV DX,OFFSET M5 INT 21H

MOV DI,0 MOV CX,NUM

MOV AH,0H

L7: MOV AL,RESULT[DI] INC DI

CALL DISPH LOOP L7

.EXIT

DISPH PROC NEAR

PUSH AX PUSH CX MOV CL,4 MOV CH,4

DISPH1:

ROL AX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP CX POP BX RET

DISPH ENDP END

# Subtract two arrays

.MODEL SMALL

.386

.DATA

ARRAY1 DB 50 DUP(?) ARRAY2 DB 50 DUP(?) RESULT DB 50 DUP(?)

M0 DB 10,13,'PROGRAM TO ADD 2 ARRAYS$' M1 DB 10,13,'ENTER THE SIZE OF ARRAYS$' M2 DB 10,13,'ENTER DIGIT$'

M3 DB 10,13,'ENTER ELEMENT OF ARRAY1$' M4 DB 10,13,'ENTER ELEMENT OF ARRAY2$' M5 DB 10,13,'RESULTING ARRAY$'

NUM DW 0H

.CODE

.STARTUP

MOV AX,0H MOV CX,0H MOV DI,0H MOV BX,0H

MOV AH,9

MOV DX,OFFSET M0 INT 21H

MOV AH,9

MOV DX,OFFSET M1 INT 21H

MOV AH,1 INT 21H SUB AL,30H MOV AH,0

MOV NUM,AX MOV DI,0H MOV CX,NUM

MOV AH,9

MOV DX,OFFSET M3 INT 21H

L1:MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV ARRAY1[DI],AL INC DI

LOOP L1

MOV DI,0 MOV CX,NUM

MOV AH,9

MOV DX,OFFSET M4 INT 21H

L2:MOV AH,9

MOV DX,OFFSET M2 INT 21H

MOV AH,1 INT 21H SUB AL,30H

MOV ARRAY2[DI],AL INC DI

LOOP L2

MOV DI,0 MOV CX,NUM

L3:

MOV DL,ARRAY1[DI] SUB DL,ARRAY2[DI] MOV RESULT[DI],DL INC DI

LOOP L3

MOV AH,9

MOV DX,OFFSET M5 INT 21H

MOV DI,0 MOV CX,NUM

MOV AH,0H

L7: MOV AL,RESULT[DI] INC DI

CALL DISPH LOOP L7

.EXIT

DISPH PROC NEAR

PUSH AX PUSH CX MOV CL,4 MOV CH,4

DISPH1:

ROL AX,CL PUSH EAX AND AL,0FH ADD AL,30H CMP AL,'9' JBE DISPH2 ADD AL,7

DISPH2:

MOV AH,2 MOV DL,AL INT 21H POP EAX DEC CH JNZ DISPH1 POP CX POP BX RET

DISPH ENDP END

# Q6 Binary to ASCII

.MODEL SMALL

.DATA

INPUT DB 10,13 , 'ENTER 8-BIT BINARY DATA : $' OUTPUT DB 10,13, 'THE ASCII CHARACTER IS : $'

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET INPUT INT 21H

MOV BL,00H MOV CL,08H

INPUT1:

MOV AH,01H INT 21H

SUB AL,30H SHL BL,1 ADD BL,AL LOOP INPUT1

MOV AH,09H

MOV DX,OFFSET OUTPUT INT 21H

MOV AH,02H MOV DL,BL INT 21H

.EXIT END

# Q7 ASCII to Binary

.MODEL SMALL

.386

.DATA

M1 DB 10,13,'ENTER ANY ASCII CHARACTER : $' M2 DB 10,13,'8-BIT BINARY RESULT IS : $'

.CODE

.STARTUP

MOV BX,0

MOV DX,OFFSET M1 MOV AH,9

INT 21H

MOV AH,1 INT 21H

MOV BL,AL

MOV DX,OFFSET M2 MOV AH,9

INT 21H

MOV CX,8 L2:

ROL BL,1 MOV DL,BL AND DL,01H ADD DL,30H MOV AH,2 INT 21H LOOP L2

.EXIT END

# Q8 Sorting (Bubble Sort)

.MODEL SMALL

.DATA A DW ?

STRING DB 10,13,'ENTER LENGTH OF ARRAY: $' STRI DB 10,13,'ENTER VALUES: $'

STR1 DB 10,13,'BUBBLE SORTED: $' ARRAY DB 10 DUP(0)

.CODE

.STARTUP MOV AH,9

LEA DX,STRING INT 21H

MOV AH,1 INT 21H SUB AL,30H AND AH,00 MOV A,AX MOV CX,A

LEA DX,STRI INT 21H

MOV BX,OFFSET ARRAY MOV AH,1

INPUTS:

INT 21H

MOV [BX],AL INC BX

LOOP INPUTS

MOV CX,A DEC CX

NEXTSCAN:

MOV BX,CX MOV SI,0

NEXTCOMP:

MOV AL,ARRAY[SI] MOV DL,ARRAY[SI+1] CMP AL,DL

JC NOSWAP

MOV ARRAY[SI],DL MOV ARRAY[SI+1],AL

NOSWAP:

INC SI DEC BX

JNZ NEXTCOMP LOOP NEXTSCAN

MOV AH,9 LEA DX,STR1 INT 21H

MOV CX,A

MOV BX,OFFSET ARRAY

PRINT:

MOV AH,2

MOV DL,[BX] INT 21H

INC BX LOOP PRINT

RET

MOV AH,4CH INT 21H

END

Microprocessor, Section A B.Sc (Hons.) Computer Science