**TASK01**

.MODEL SMALL

.STACK 100H

.DATA

; DEFINE YOUR VARIABLES HERE

X DB "WHAT IS THE LENGTH OF YOUR NAME(including space) $"

ARR DB 50 DUP(?) ; declaring array with null value

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

SUB BX,BX

SUB CX,CX

; YOUR CODE STARTS HERE

LEA DX,X

MOV AH,9

INT 21h

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV DL,10

MOV BL,0

AGAIN:

MOV AH,01H

INT 21H

CMP AL,13

JE NEXT

MOV AH,0

SUB AL,48

MOV CL,AL

MOV AL,BL

MUL DL

ADD AL,CL

MOV BL,AL

JMP AGAIN

NEXT:

CMP BL,15

JLE STEP1

JG STEP2

STEP1:

MOV AL,BL

MOV BX,AX

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV AX,BX

MOV CL,AL

MOV BL,AL

MOV SI,0

INPUT:

INT 21H

MOV ARR[SI],AL

INC SI

LOOP INPUT

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV CX,BX

MOV SI,0

MOV AH,2

OUTPUT:

MOV DL,ARR[SI]

MOV DH,41H

CMP DL,61H

JGE SMALL

CMP DL,20H

JE SPACE

JMP CAPITAL

SMALL:

CMP DL,7AH

JLE S1

JMP CAPITAL

S1:

MOV DL,68H

INT 21h

INC SI

LOOP OUTPUT

JMP DONE

CAPITAL:

CMP DL,41H

JGE C1

JMP DONE

C1:

CMP DL,5AH

JLE C2

JMP DONE

C2:

MOV DL,47H

INT 21h

INC SI

LOOP OUTPUT

SPACE:

CMP DL,20H

JE L1

JMP DONE

L1:

INT 21h

INC SI

LOOP OUTPUT

STEP2:

ADD BL,6H

MOV AL,BL

MOV BX,AX

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV AX,BX

MOV CL,AL

MOV BL,AL

SUB CL,6H

INPUT1:

INT 21H

MOV ARR[SI],AL

INC SI

LOOP INPUT1

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV CX,BX

MOV SI,0

MOV AH,2

OUTPUT1:

MOV DL,ARR[SI]

MOV DH,41H

CMP DL,61H

JGE SMALL1

CMP DL,20H

JE SPACE1

JMP CAPITAL1

SMALL1:

CMP DL,7AH

JLE S6

JMP CAPITAL1

S6:

MOV DL,68H

INT 21h

INC SI

LOOP OUTPUT1

JMP DONE

CAPITAL1:

CMP DL,41H

JGE C6

JMP DONE

C6:

CMP DL,5AH

JLE C7

JMP DONE

C7:

MOV DL,47H

INT 21h

INC SI

LOOP OUTPUT1

SPACE1:

CMP DL,20H

JE L4

JMP DONE

L4:

INT 21h

INC SI

LOOP OUTPUT1

DONE:

; YOUR CODE ENDS HERE

MOV AX, 4C00H

INT 21H

MAIN ENDP

END MAIN

**TASK02**

.MODEL SMALL

.STACK 100H

.DATA

; DEFINE YOUR VARIABLES HERE

W DB "Enter the four numbers(press enter after entering each 8 bit input): $"

ARR DB ?

X DB "Sum: $"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV SI,OFFSET ARR

; YOUR CODE STARTS HERE

LEA DX,W ;SHOWING THE PROMPT

MOV AH,9

INT 21h

MOV CX,4 ;SETTING THE COUNTER TO 4

INPUT: ;TAKING INPUTS

MOV AH,2 ;GOING TO NEXT LINE

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV DL,10

MOV BL,0

AGAIN:

MOV AH,01H

INT 21H

CMP AL,13

JE NEXT

MOV AH,0

SUB AL,48

MOV BH,AL

MOV AL,BL

MUL DL

ADD AL,BH

MOV BL,AL

JMP AGAIN

NEXT:

MOV AL,BL

MOV [SI],AL

INC SI

LOOP INPUT

MOV AH,2 ;GOING TO NEXT LINE

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

MOV SI,OFFSET ARR ;PRINTING THE ARRAY

MOV CX,4

SUB BX,BX

OUTPUT:

MOV DL,[SI]

;IDENTIFYING ODD OR EVEN

MOV AX,DX

MOV BL,2H

DIV BL

CMP AH,0H

JNE L1

INC SI

LOOP OUTPUT

JMP DONE

L1:

CMP BH,0H

JE L2

JNE L3

JMP DONE

L2: ;STORING FIRST ODD NUMBER

ADD BH,DL

INC SI

LOOP OUTPUT

L3:

ADD CH,DL

SUB DX,DX

MOV BL,BH

SUB BH,BH

MOV CL,CH

SUB CH,CH

ADD BX,CX

JMP DONE

DONE:

MOV AX,BX

MOV CX,0

MOV DX,0

LABEL1:

CMP AX,0

JE PRINT1

MOV BX,10

DIV BX

PUSH DX

INC CX

XOR DX,DX

JMP LABEL1

PRINT1:

CMP CX,0

JE EXIT

POP DX

ADD DX,48

MOV AH,02h

INT 21h

DEC CX

JMP PRINT1

EXIT:

; YOUR CODE ENDS HERE

MOV AX, 4C00H

INT 21H

MAIN ENDP

END MAIN

**TASK03**

.MODEL SMALL

.STACK 100H

.DATA

; DEFINE YOUR VARIABLES HERE

ARR DB 8,6,2,9,5

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; YOUR CODE ENDS HERE

MOV DL,0

L1:

CMP DL,3

JG PRINT

MOV CX,4

MOV SI,0

MOV DI,1

L2:

MOV AL,ARR[SI]

MOV BL,ARR[DI]

CMP AL,BL

JG L3 ;IF AL IS LARGE

JMP L4

L3:

MOV ARR[SI],BL ;STORING SMALL ONE

MOV ARR[di],AL ;STORING LARGE ONE

L4:

ADD SI,1

ADD DI,1

LOOP L2

ADD DL,1 ;SORTING COMPLETED

JMP L1

PRINT:

MOV CX,5 ;PRITING COMMAND STARTED FROM HERE

MOV SI,0

MOV AH,2

DONE:

MOV DL,ARR[SI]

ADD DL,30H

INT 21H

MOV DL,20H

INT 21H

ADD SI,1

LOOP DONE

; YOUR CODE ENDS HERE

MOV AX, 4C00H

INT 21H

MAIN ENDP

END MAIN