.MODEL SMALL

.STACK 100H

.DATA

; DEFINE YOUR VARIABLES

MSG DB "ENTER 4 DIGIT NUMBER: $"

MSG1 DB "No Duplicates $"

A DW ?

B DW ?

C DB ?

D DW ?

X DW ?

Y DW ?

ARRAY DB 50 DUP(?)

SAME DB 50 DUP(?)

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; YOUR CODE STARTS HERE

;TAKING INPUT FROM USER

LOOP1:

CMP C, 2

JGE DONE

;PRINTING THE MSG

LEA DX, MSG

MOV AH, 9

INT 21h

MOV BH, 0

;LOOP STARTING

LOOP2:

CMP BH, 4 ;SETTING BH=4 AS I NEED 4 DIGIT NUMBER

JGE T1 ;BH<=4

MOV AH, 1 ;TAKING INPUT

INT 21H

SUB AH,AH ;CLEARING AH

SUB AL,48 ;GETTING THE NUMBER IN HEX

PUSH AX ;KEEPING IT IN STACK

INC BH ;BH+=1

JMP LOOP2

T1:

MOV AH, 2 ;RETURNING

MOV DL, 0Dh

INT 21h

MOV DL,0AH

INT 21h

MOV BH, 0

LOOP3:

CMP BH, 4 ;BH=4 AS LOOP WILL CONTINUE FOR 4 TIMES

JGE ENDING

POP DX ;POPS THE LAST ONE FROM STACK

INC BH ;BH+=1

MOV A, DX ;A=DX

MOV CX, 10

MOV AX, A

MOV bl, 1

LOOP4:

CMP BL, BH

JGE LOOP5

MUL CX

INC BL

JMP LOOP4

LOOP5:

ADD B, ax

JMP LOOP3

ENDING:

CMP C, 0

JE LOOP6

JMP LOOP7

LOOP6:

MOV CX, B ;WHEN G=0 CX=B

MOV B, 0 ;CLEARING B

MOV D, CX ;D=CX

JMP LOOP7

LOOP7:

INC C ;C+=1

JMP LOOP1

DONE:

;FIRST INPUT IS STORED IN D

;SECOND INPUT IS STORED IN B

;STORING FIRST INPUT IN ARRAY

MOV AX,D ;MOVING THE VALUE OF D IN AX

MOV BX,3E8H ;BX=1000

DIV BX ;D/BX

MOV ARRAY[0],AL ;SAVING THE ANSWER AFTER DIVIDING

MOV AX,DX ;AX=DX

SUB DX,DX ;CLEARING DX

MOV BX,0064H ;BX=100

DIV BX ;AX/BX

MOV ARRAY[1],AL ;SAVING THE ANSWER AFTER DIVIDING

MOV AX,DX ;AX=DX

SUB DX,DX ;CLEARING DX

MOV BX,000AH ;BX=10

DIV BX ;AX/BX

MOV ARRAY[2],AL ;SAVING THE ANSWER AFTER DIVIDING

MOV AX,DX ;AX=DX

MOV ARRAY[3],AL ;SAVING THE LAST REMAINING VALUE

SUB AX,AX ;CLEARING AX

SUB BX,BX ;CLEARING BX

SUB CX,CX ;CLEARING CX

SUB DX,DX ;CLEARING DX

;STORING SECOND INPUT IN ARRAY

MOV AX,B ;MOVING THE VALUE OF B IN AX

MOV BX,3E8H

DIV BX

MOV ARRAY[4],AL

MOV AX,DX

SUB DX,DX

MOV BX,0064H

DIV BX

MOV ARRAY[5],AL

MOV AX,DX

SUB DX,DX

MOV BX,000AH

DIV BX

MOV ARRAY[6],AL

MOV AX,DX

MOV ARRAY[7],AL

;STORING VALUE IN THE ARRAY IS COMPLETE

;PRINTING THE ARRAY

MOV SI,OFFSET ARRAY ;SI=OFFSET OF THE ARRAY

MOV CX,8 ;CX=8 SO THE LOOP WILL CONTINUE FOR 8 TIMES

OUTPUT:

MOV DL,[SI] ;DL IS HOLDING ARRAY ELEMENT

ADD DL,30H ;DL+=30H

MOV AH,2 ;PRINTING COMMAND

INT 21H

INC SI ;SI+=1

MOV DL,20H ;PRINTING SPACE BETWEEN TWO NUMBERS

MOV AH,2

INT 21H

LOOP OUTPUT

;PRINTING ARRAY IS COMPLETED

;GOING NEXT LINE

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

;REACHED NEXT LINE

;CONVERTING 0 TO RANDOM VALUE(62H)

MOV BH,ARRAY[0] ;BH=ARRAY[0]

CMP BH,0H ;IF BH=0 THEN GO TO Z1 ELSE X1

JE Z1

JNE X1

Z1:

MOV ARRAY[0],62H ;ARRAY[0]=b

X1:

MOV BH,ARRAY[1] ;BH=ARRAY[1]

CMP BH,0H ;IF BH=0 THEN GO TO Z2 ELSE X2

JE Z2

JNE X2

Z2:

MOV ARRAY[1],62H ;ARRAY[1]=b

X2:

MOV BH,ARRAY[2] ;BH=ARRAY[2]

CMP BH,0H ;IF BH=0 THEN GO TO Z3 ELSE X3

JE Z3

JNE X3

Z3:

MOV ARRAY[2],62H ;ARRAY[2]=b

X3:

MOV BH,ARRAY[3] ;BH=ARRAY[3]

CMP BH,0H ;IF BH=0 THEN GO TO Z4 ELSE X4

JE Z4

JNE X4

Z4:

MOV ARRAY[3],62H ;ARRAY[3]=b

X4:

MOV BH,ARRAY[4] ;BH=ARRAY[4]

CMP BH,0H ;IF BH=0 THEN GO TO Z5 ELSE X5

JE Z5

JNE X5

Z5:

MOV ARRAY[4],62H ;ARRAY[4]=b

X5:

MOV BH,ARRAY[5] ;BH=ARRAY[5]

CMP BH,0H ;IF BH=0 THEN GO TO Z6 ELSE X6

JE Z6

JNE X6

Z6:

MOV ARRAY[5],62H ;ARRAY[5]=b

X6:

MOV BH,ARRAY[6] ;BH=ARRAY[6]

CMP BH,0H ;IF BH=0 THEN GO TO Z7 ELSE X7

JE Z7

JNE X7

Z7:

MOV ARRAY[6],62H ;ARRAY[6]=b

X7:

MOV BH,ARRAY[7] ;BH=ARRAY[7]

CMP BH,0H ;IF BH=0 THEN GO TO Z8 ELSE X8

JE Z8

JNE X8

Z8:

MOV ARRAY[7],62H ;ARRAY[7]=b

;PRINTING ARRAY IS COMPLETED

X8:

;TRAVERSING

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,0H

;COMPARING ALL ELEMENTS WITH FIRST ELEMENT

TRAVERSING1:

MOV CX,8 ;CX=8

MOV BH,ARRAY[0] ;BH=ARRAY[0]

L1:

INC SI ;SI+=1

MOV DL,ARRAY[SI] ;DL=ARRAY[SI]

CMP BH,DL ;IF BH=DL GO TO STEP L2 OTHERWISE LOOP L1 WILL CONTINUE UNTIL COMPARING IS DONE

JE L2

LOOP L1

JMP TRAVERSING2

L2:

MOV SAME[0],BH ;AS COMPARED VALUE IS SAME WITH THE FIRST ELEMENT SO ARRAY[0]=BH

MOV ARRAY[SI],0H ;DELETING COMPARED ELEMENT

LOOP L1:

;COMPARING ALL ELEMENTS WITH SECOND ELEMENT

TRAVERSING2:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,1H

MOV CX,8

MOV BH,ARRAY[1]

L3:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L4

LOOP L3

JMP TRAVERSING3

L4:

MOV SAME[1],BH

MOV ARRAY[SI],0H

LOOP L3:

;COMPARING ALL ELEMENTS WITH THIRD ELEMENT

TRAVERSING3:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,2H

MOV CX,8

MOV BH,ARRAY[2]

L5:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L6

LOOP L5

JMP TRAVERSING4

L6:

MOV SAME[2],BH

MOV ARRAY[SI],0H

LOOP L5:

;COMPARING ALL ELEMENTS WITH FOURTH ELEMENT

TRAVERSING4:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,3H

MOV CX,8

MOV BH,ARRAY[3]

L7:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L8

LOOP L7

JMP TRAVERSING5

L8:

MOV SAME[3],BH

MOV ARRAY[SI],0H

LOOP L7:

;COMPARING ALL ELEMENTS WITH FIFTH ELEMENT

TRAVERSING5:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,4H

MOV CX,8

MOV BH,ARRAY[4]

L9:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L10

LOOP L9

JMP TRAVERSING6

L10:

MOV SAME[4],BH

MOV ARRAY[SI],0H

LOOP L9:

;COMPARING ALL ELEMENTS WITH SIXTH ELEMENT

TRAVERSING6:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,5H

MOV CX,8

MOV BH,ARRAY[5]

L11:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L12

LOOP L11

JMP TRAVERSING7

L12:

MOV SAME[5],BH

MOV ARRAY[SI],0H

LOOP L11:

;COMPARING ALL ELEMENTS WITH SEVENTH ELEMENT

TRAVERSING7:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,6H

MOV CX,8

MOV BH,ARRAY[6]

L13:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L14

LOOP L13

JMP TRAVERSING8

L14:

MOV SAME[6],BH

MOV ARRAY[SI],0H

LOOP L13:

;COMPARING ALL ELEMENTS WITH LAST ELEMENT

TRAVERSING8:

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

MOV SI,7H

MOV CX,8

MOV BH,ARRAY[7]

L15:

INC SI

MOV DL,ARRAY[SI]

CMP BH,DL

JE L16

LOOP L15

JMP FINISHED

L16:

MOV SAME[7],BH

MOV ARRAY[SI],0H

LOOP L15:

FINISHED:

;PRINTING COMMON ELEMENTS

MOV SI,7 ;SAME.LENGTH=7

MOV CX,8 ;LOOP WILL CONTINUE FOR 8 TIMES

OUTPUT1:

MOV DL,SAME[SI] ;DL IS HOLDING ARRAY ELEMENT

ADD DL,30H ;DL+=30

CMP DL,30H ;IF DL=0 THEN GOING TO S2 ELSE S1

JNE S1

JE S2

S1: ;DL!=0

CMP DL,92H ;IF DL=THAT RANDOM VALUE WHICH INDICATES 0 ACTUALLY THEN 4 ELSE S5

JE S4

JNE S5

S5:

MOV AH,2 ;GOING NEXT LINE

INT 21H

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH ;PRINTING INDEX

INT 21h

JMP S2

S4:

MOV DL,30H

MOV AH,2

INT 21H

MOV AH,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

S2:

DEC SI

LOOP OUTPUT1

SUB AX,AX

SUB BX,BX

SUB CX,CX

SUB DX,DX

;EMPTY ARRAY PRINTING

MOV BH,SAME[0]

CMP BH,0H

JE B1

JNE COMPLETED

B1:

MOV BH,SAME[1]

CMP BH,0

JE B2

JNE COMPLETED

B2:

MOV BH,SAME[2]

CMP BH,0

JE B3

JNE COMPLETED

B3:

MOV BH,SAME[3]

CMP BH,0

JE B4

JNE COMPLETED

B4:

MOV BH,SAME[4]

CMP BH,0

JE B5

JNE COMPLETED

B5:

MOV BH,SAME[5]

CMP BH,0

JE B6

JNE COMPLETED

B6:

MOV BH,SAME[6]

CMP BH,0

JE B7

JNE COMPLETED

B7:

MOV BH,SAME[7]

CMP BH,0

JE B8

JNE COMPLETED

B8:

LEA DX, MSG1

MOV AH, 9

INT 21h

COMPLETED:

; YOUR CODE ENDS HERE

MOV AX, 4C00H

INT 21H

MAIN ENDP

END MAIN