

```

.MODEL SMALL
.STACK 100H
.DATA
    MSGA DB 'Number of Inputs(max 5 digits): $'
    MSGB DB 'Enter Number: $'
    MSGC DB 'Maximum: $'
    MSGD DB 'Minimum: $'
    MAX DW 0
    MIN DW 9999
    TEMP DW ?
    COUNT DW ?

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    LEA DX, MSGA                ;Print 'Number of Inputs:'
    MOV AH,9
    INT 21H

GET_INPUT:
    MOV AH,1
    MOV BX,0
    INT 21H

    CMP AL,0DH
    JE END_INPUT                ;If Enter

    INNER_LOOP_1:
        MOV AH,0                ;Use full 16 bits of AX
        SUB AX,48

        MOV TEMP,AX
        MOV AX,10
        MUL BX                    ;AX = AX*BX
        MOV BX,AX
        ADD BX,TEMP

    MOV AH,1                    ;Input new digit
    INT 21H
    CMP AL,0DH
    JNE INNER_LOOP_1            ;If Enter

END_INPUT:
    MOV COUNT,BX

    LEA DX, MSGB                ;Print 'Enter Number'
    MOV AH,9
    INT 21H

    MOV AH,2
    MOV DL,0DH
    INT 21H                    ;New line
    MOV DL,0AH
    INT 21H

    MOV DX,0
    MOV TEMP,0

OUT_LOOP:
    MOV AH,1
    MOV BX,0
    INT 21H

    CMP AL,0DH

```

```

JE LAST_INPUT                ;If Enter

INNER_LOOP_2:
    MOV AH,0                 ;Use full 16 bits of AX
    SUB AX,48

    MOV TEMP,AX
    MOV AX,10
    MUL BX                   ;AX = AX*BX
    MOV BX,AX
    ADD BX,TEMP

    MOV AH,1                 ;Input new digit
    INT 21H
    CMP AL,0DH
    JNE INNER_LOOP_2        ;If Enter

LAST_INPUT:
    CMP BX,MAX
    JBE CHECK_MIN
    MOV MAX,BX

CHECK_MIN:
    CMP BX,MIN
    JAE LINK_UP
    MOV MIN,BX

LINK_UP:
    DEC COUNT
    CMP COUNT,0
    JNE OUT_LOOP

PRINT:
    LEA DX, MSGC             ;Print 'Maximum:'
    MOV AH,9
    INT 21H

    MOV CX,0                 ;Or XOR CX,CX
    MOV AX,MAX
    MOV BX,10

STOR_MAX:
    MOV DX,0
    DIV BX                   ;AX = AX/BX
    PUSH DX                  ;Reminder(DX) in stack
    INC CX
    CMP AX,0
    JNE STOR_MAX

PRINT_MAX:
    MOV AH,2
    POP DX
    ADD DL,48                 ;Pop stack by DX and DX = DL
    INT 21H

    DEC CX
    CMP CX,0
    JNZ PRINT_MAX

    MOV DL,0DH
    INT 21H                   ;New line
    MOV DL,0AH
    INT 21H

NEXT_STEP:
    LEA DX, MSGD             ;Print 'Minimum:'

```

```
MOV AH,9
INT 21H

MOV CX,0                ;Or XOR CX,CX
MOV AX,MIN
MOV BX,10
```

STOR_MIN:

```
MOV DX,0
DIV BX                ;AX = AX/BX
PUSH DX              ;Reminder(DX) in stack
INC CX
CMP AX,0
JNE STOR_MIN
```

PRINT_MIN:

```
MOV AH,2
POP DX                ;Pop stack by DX and DX = DL
ADD DL,48
INT 21H

DEC CX
CMP CX,0
JNZ PRINT_MIN
```

END_PROC:

```
MOV AH,4CH
INT 21H
```

MAIN ENDP

```
END MAIN
```