8-bit addition

```
DATA SEGMENT
NO1 db 05H
NO2 db 07H
resl db ?
resh db ?
DATA ENDS

ASSUME DS: DATA, CS: CODE
CODE SEGMENT
START:

MOV AX, DATA
MOV DS, AX

MOV AL, NO1
MOV BL, NO2
ADD AL, BL
MOV resl, AL
MOV resh, 0000H
JNC SKIP
MOV resh, 0001H
SKIP: INT 03H

CODE ENDS
END START
```

16-bit addition

```
DATA SEGMENT
NO1 dw 4567H
NO2 dw 1652H
resl dw ?
resh dw ?
DATA ENDS

ASSUME DS: DATA, CS: CODE
CODE SEGMENT
START:

MOV AX, DATA
MOV DS, AX

MOV AX, NO1
MOV BX, NO2
ADD AX, BX
MOV resl, AX
MOV resh, 0000H
JNC SKIP
MOV resh, 0001H
SKIP: INT 03H

CODE ENDS
END START
```

#ASSENDING

```
DATA SEGMENT
STRING1 DB 99H,12H,56H,45H,36H
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
MOV DS,AX

MOV CH,04H

UP2: MOV CL,04H
LEA SI,STRING1

UP1: MOV AL,[SI]
MOV BL,[SI+1]
CMP AL,BL
JC DOWN
MOV DL,[SI+1]
XCHG [SI],DL
MOV [SI+1],DL

DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2

INT 3
CODE ENDS
END START
```

#DESENDING

```
DATA SEGMENT
STRING1 DB 99H,12H,56H,45H,36H
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
MOV DS,AX

MOV CH,04H

UP2: MOV CL,04H
LEA SI,STRING1

UP1:MOV AL,[SI]
MOV BL,[SI+1]
CMP AL,BL
JNC DOWN
MOV DL,[SI+1]
XCHG [SI],DL
MOV [SI+1],DL

DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2

INT 3
CODE ENDS
END START
```

#FACTORIAL:

```
DATA SEGMENT
STRING DB 3
DATA ENDS

ASSUME DS: DATA, CS: CODE
CODE SEGMENT
START:
MOV AX, DATA
MOV DS, AX

MOV AH,00
MOV AL, STRING

L1:  DEC STRING
MUL STRING
MOV CL, STRING

CMP CL,01
JNZ L1
MOV AH,4CH
INT 21H

CODE ENDS
END START
```