Add two 16-bit numbers using different addressing mode and show result with status flags.

```
mov ax, 1234h      ; immediate
mov bx, 1111h      ; immediate
add ax, bx         ; register addressing
```

Subtract two 16-bit numbers using different addressing mode and show result with status flags.

```
mov ax,500h;
mov bx,123h;
sub ax,bx;
```

Multiply two numbers and show result with status flags.

```
mov ax,200h
mov bx,20h
mul bx
hlt
```

Perform division of two numbers and show result with status flags.

```
mov ax,200h;
mov bx,21h;
div bx
hlt;
```

Add all numbers from 1 to 10 and display result with status flags.

```
mov ax,0;
mov bx,1;
mov cx,10;
start:
   add ax,bx;
   add bx,1;
loop start
hlt;
```

Add all odd numbers from 1 to 10 and display result with status flags.

```
mov ax,1;
mov bx,0;
mov dx,0;
mov cx,5;
start:
   add dx,ax;
   add ax,bx;
   add bx,2;
loop start;
hlt;
```

Add all even numbers from 1 to 10 and display result with status flags.

```
mov ax,2;
mov bx,0;
mov dx,0;
mov cx,5;
start:
   add dx,ax;
   add ax,bx;
   add bx,2;
loop start;
hlt;
```

Perform AND / OR / NOT / XOR operation of two numbers and display result with status flags.

```
mov ax,10h;
mov bx,20h;
mov dx,ax
and dx,bx ; and in ax*bx
;----------------------
mov dx,ax;
or dx,bx ;
;----------------------
mov dx,ax;
not dx ; not ax
;-----------------------
mov dx,ax;
xor dx,bx; xor in ax^bx
```

Perform different shift operation and show the result with status flags.

```
mov ax, 7fa4h;
mov dx,ax
mov cl,02h;
shl dx,cl; logical left shift
;----------------------------
mov dx,ax;
sal dx,cl; arithmetic shift
;----------------------------
mov dx,ax;
shr dx,cl; logical right shift
sar dx,cl; arithmetic right shift
hlt;
```

Take input from an input port and send the data to an output port.

```
; input
mov ah,01h ; read 1key
int 21h ; for dos
; input will be saved in AL (in hex)
; to show it in decimal ;

; output
mov dl,al;
mov ah,02h ;print 1key
int 21h
```