Q1:

```
[org 0x100]

jmp start
num1 : dw 5
num2 : dw 24
num3 : dw 0


start :
mov ax , 5
mov cx , 10
mov bx , 24
mov bi , [num3]
mov dx , 2

l1 :
add dx , dx
sub ax ,1
jne l1

xor bx , dx

mov bi , 0

l2:
add dx , dx
sub cx , 1
jne l2

not dx
xor bx , dx

mov ax , 0x4c00
int 0x21
```

Q2:

```
[org 0x100]

jmp start :
num1 : 24


start:
mov bx , num1
mov dx , 1
mov ax , 0
mov si , 16
l1:
test bx , dx
jc l2
sub si , 1
cmp si , 0
```

```
je end


l2 :
add ax , 1
add dx, dx
jmp l1

end :

mov ax , 0x4c00
int 0x21

Q3:

[org 0x100]


mov si, -2
mov di, 10
mom

bx, numi
11:
sub di, 2
add si,2
mov ax, [numi+sil
 cmp ax, [num2 +di]
 mov dx,1
je 11
 mov dx, 0
 mov ax, 6x4c00
 int 21h

num1; dw 1,2,3,2,1

Q4:

[org 0x0100]



jmp start
num : dw 5

start :

mov ax,0

 mov dx,[num]  ; this is the X
 mov bx,[num]  ; this is for the expression [x-1]
 dec bx
 mov cx,bx     ; counter
```

```
l1:
add ax,dx
sub cx,1
jz subsitution
cmp cx,0
jne l1

 subsitution:
 mov dx,ax
 mov ax,0
 dec bx
 mov cx,bx
 cmp cx,1
 jnz l1


 mov ax,0x4C00
 int 21h
```