Q1:

```
org 100h

push arr
push sizearr
push word [a]
push word [l]
push word [r]
call xorq
mov ah,4ch
int 21h


xorq:
push bp
mov bp,sp


push bx
push cx
push si


mov ax,[bp+8]
mov bx,[bp+12]
mov si,[bp+6]
mov cx,[bp+4]
sub cx,si

push cx
push bx
push si

doit:
xor [bx+si], ax
add si,2
loop doit

pop si
pop bx
pop cx

mov ax,[bx+si]
add si,2

doit2:
cmp ax,[bx+si]
ja toloop
mov ax,[bx+si]
toloop:
add si,2
loop doit2

pop si
pop cx
```

```
pop bx


pop bp
ret 6


arr: dw 3,5,9
sizearr: dw 7
a: dw 4
l: dw 0
r: dw 2



Q2:
[org 0x0100]

jmp start

Array1: db 1,2,3,4,5,6,7
Array2: db 9,6,4,1
Size1: db 7
Size2: db 4


start:
 mov si, Array1
 mov ax, 0
 mov al, byte[Size1]
 add si, ax
 sub si,1
 mov di, Array1


 doagin:
  mov al, byte[si]
  mov bl, byte[di]
  mov byte[si], bl
  mov byte[di], al
  dec si
  inc di
  cmp di, si
  jb doagin

 mov si, Array1
 mov di, Array2

 actually:
  mov al, byte[si]
  cmp al, [di]
  jb shift
  inc si
  jmp actually
```

```
 shift:
  mov bh, byte[di]

  mov ax, si
  mov dx, Array1
  sub ax, dx
  xor cx, cx
  mov cl, byte[Size1]
  sub cx, ax
  mov bp, 0
  mov bp, di
  mov ax, Array2
  sub bp, ax
  add bp, cx

 hereLoops:

  mov bl, byte[si+bp-1]
  mov byte[si+bp], bl
  dec bp
  jnz hereLoops

  mov byte[si], bh
  inc di
  inc si

  mov ax, Array2
  mov bl, byte[Size2]
  mov bh, 0
  add ax, bx

  mov bx, Array1
  mov cl, byte[Size1]
  mov ch, 0
  add bx, cx
  cmp si, bx
  jz EXIT
  cmp di, ax
  jnz actually

EXIT:
mov ax, 0x4c00 ; terminate program
   int 0x21
```