```
Question 1
[org 0x0100]
        jmp start

data:   dw      60, 40, 50, 22, 5
data2:  db      3, 4, 1, 9, 6

swapflag: db    0

swap:
        cmp     dx,1
        jne     l4
        mov     ax, [bx+si]
        xchg    ax, [bx+si+2]
        mov     [bx+si], ax
        jmp     l4a
l4:
        mov     al, [bx+si]
        xchg    al, [bx+si+1]
        mov     [bx+si], al
l4a:
        ret
bubblesort:
        push    ax
        push    cx
        push    si
        push    dx

        dec     cx
        cmp     dx,1
        jne     mainloop
        shl     cx, 1

mainloop:
        mov     si, 0
        mov     byte [swapflag], 0

innerloop:
        cmp     dx,1
        jne     l1
        mov     ax,[bx+si]
        cmp     ax,[bx+si+2]
        jmp     l1a
l1:
        mov     al,[bx+si]
        cmp     al,[bx+si+1]
l1a:
        jbe     noswap

        call    swap
        mov     byte[swapflag],1

noswap:
        cmp     dx,1
        jne     l2
        add     si, 1
l2:
        add     si, 1
```

```
        cmp     si, cx
        jne     innerloop

        cmp     byte [swapflag], 0
        je      done
        cmp     dx,1
        jne     l3
        sub     cx,1
l3:
        sub     cx,1
        jnz     mainloop

done:
        pop     dx
        pop     si
        pop     cx
        pop     ax

        ret

start:
        mov     bx, data
        mov     cx, 5
        mov     dx,1
        call    bubblesort

        mov     bx, data2
        mov     cx, 5
        mov     dx,0
        call    bubblesort

        mov     ax, 0x4c00
        int     0x21

Question 2
[org 0x0100]
        jmp     start
fib:
        push    bp
        mov     bp, sp
        sub     sp, 4
        mov     word [bp-2], di;        value of x
        cmp     word [bp-2], 1
        jne     l1
        mov     ax, word [bp-2]
        jmp     l2
l1:
        cmp     word [bp-2], 0
        jne     l3
        mov     ax, word [bp-2]
        jmp     l2
l3:
        mov     ax, word [bp-2]
        sub     ax, 1
        mov     di, ax
        call    fib
        mov     word [bp-4], ax
        mov     ax, word [bp-2]
```

```
        sub     ax, 2
        mov     di, ax
        call    fib
        add     ax, word [bp-4]
l2:
        mov     sp,bp
        pop     bp
        ret

start:
        mov     di,0
        call    fib

        mov     di,1
        call    fib

        mov     di,2
        call    fib

        mov     di,3
        call    fib

        mov     di,4
        call    fib

        mov     di,5
        call    fib

        mov     di,6
        call    fib

        mov     di,7
        call    fib

        mov     di,8
        call    fib

        mov     di,9
        call    fib

        mov     di,10
        call    fib

        mov     ax,0x4c00
        int     0x21
```