2.

a)

For some integer Array A[1, …, n] where n is odd, the while loop will execute (n-1)/2 times. And within each while loop, 3 comparisons will be performed. And in the end, the while loop will execute one last time in order to quit.

So, the total number of comparisons Y, as a function of n, will be

$$Y = (n-1) / 2 * 3 + (n - 1) / 2 + 1$$

$$= \frac{3n-3}{2} + \frac{n-1}{2} + 1$$

$$= \frac{4n-4}{2} + 1$$

$$= 2n - 2 + 1$$

$$= 2n - 1$$

There are total of 2n – 1 comparisons.

b)

min ← A[1], max ← A[1]

i ← 2

while i ≤ n **do**

    if i < n **then**

        if A[i] ≤ A[i+1] **then**

            if A[i] < min **then** min ← A[i]

            if A[i+1] > max **then** max ← A[i+1]

        else

            if A[i] > max **then** max ← A[i]

            if A[i+1] < min **then** min← A[i+1]

    else

        if A[i] < min **then** min ← A[i]

        if A[i] > max **then** max ← A[i]

    i ← i + 2

return min, max

And the red parts are where I made edits.