1)

Text, letter

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

2)

**Question 1**

MaxMin(array, size):

if size == 1:

return value for max and min.

else if size == 2:

compares the two values and return larger one as max and other as min

else size > 2

divide array in two half

Recursive for max and min of left half

Recursive for max and min of right half

compare the two max elements and return the maximum as the array's max and;

Similarly, compare the two min elements and return the minimum as the array's min.

return the pair of max and min

**Question 2**

T(n) = O(n).  
We can prove it:  
T(n) = T( floor(n/2) ) + T( ceil(n/2) ) + 2 ............................ the 2 is because of the two comparisions  
Now, we have n = 2something, thus  
T(n) = 2T( n/2 ) + 2  
T(2) = 1  
T(1) = 0  
Solving this Recurion:  
T(n) = 2T(n/2) + 2  
   = 2(2(T(n/4) + 2) +2 = 2(2(2T(n/8) + 2) + 2) + 2 ...... and so on till we pass 2 in the function.  
i.e. till (n / 2m) = 2 after m steps.  
which implies that 2m+1 = n i.e. m = log2(n) - 1.  
which leads to:  
T(n) = 2mT(2) + 2m-1 + 2m-2 + .... + 2 = 2m + 2m-1 + 2m-2 + .... + 2  
=> T(n) = 2(2m - 1)/(2-1) = 2( (n / 2) - 1) = n - 2 = O(n).

3)

Text, letter

Description automatically generated