

Question 1

assume that the FindMax function can not find the maximum value in the input vector for non empty input vectors => result(the variable) is not the maximum value => there is a situation that result is not updated when n is bigger than result => contradict with result is updated to n if n is bigger than result.

Question 2

<https://github.com/guiquansun>

<https://stackoverflow.com/users/12720482/guiquan>

Question 3

I take 4 inputs to test.

Inputs = {1, 2, 3, 4}

Inputs = {1, 2, 3, 3, 4}

Inputs = {2}

Inputs = {}

All the outputs are correct. The first one is 2.5. The second one is 3. The third is 2. The last one is {}.

The time complexity is $O(1)$.

Question 4

- 3. -r : remove all the files under current directory
- f : mandatory deletion.

Question 5

1. $T(n) = n + n/2 + n/4 + \dots + 1$
 $= (1 + 1/2 + 1/4 + \dots) * n + 1$
 $= 2n + 1$

So the time complexity is $O(n)$

2. Assume the loop executes t times, so $2^t = n \Rightarrow t = \log_2^n$

So the time complexity is $O(\log n)$

Question 6

1. the result is -1 when $n < 0$. The code is in the file.
2. Yes, I can. The code is in the file.
3. take 4 inputs, $n = 5$, $n = 4$, $n = 1$, $n = 0$. All the final results are correct, 120, 24, 1 and 1.
4. The time complexity of non-recursive version is $O(n)$.
The time complexity of recursive version is also $O(n)$.
5. I try to use mathematical induction to prove the correctness of the non-recursive version.
 - a. $n = 0$, $n! = 1$, so the algorithm is correct.
 - b. $n = 1$, $n! = 1$, so the algorithm is correct.
 - c. We assume $n = k$, the algorithm is correct and $n! = m$.
When $n = k + 1$, $n! = (k + 1) * m = (k + 1)! = n!$, so the algorithm is correct.