**Problem description:**

If you have 50 boxes that contains 50 pieces of metal all of the same known weight.

one of these boxes contains fake metal pieces that weigh 1 kilogram less than the pieces in the rest of the boxes.

You can use a digital scale only once to find this fake box.

Design a brute force algorithm to solve this problem.

**Detailed assumptions:**

1) If we assume that each real box weigh **[ a kilogram ]** , then fake box weigh **[ a-1 kilograms ]**

2) Then each real metal piece in real boxes weight

3) Then each real metal piece in real boxes weight

4) we suppose in our program that real box weigh **1 kilogram**

**Detailed solution including the pseudo-code and the description of the steps of your solution:**

First step : we give each box a unique number from 1 to 50

second step : we take a number of metal pieces from each box depending on box's number

Example: 1 piece of box 1 and 2 pieces of box 2 and so on until we take 50 pieces of box 50

Third step : we weigh the gathered pieces and store the result

Fourth step : by computing 50 probabilities of expected results and storing them

Fifth step : comparing the result the digital scale with computed results, we find the fake box

findFakeBox(array boxes,float digitalScaleReading)

// boxes: array of number of pieces taken from each box

float sum = 0;

arrayOfProbablities[50]

for i <- 1 to 50

sum = 0;

for j <- 0 to 49

if j+1 == i

sum += boxes[j] \* 0.98;

else

sum += boxes[j] \* 1;

endfor

arrayOfProbablities[i-1] = sum;

endfor

for i <- 0 to 49

if digitalScaleReading == arrayOfProbablities[i]

return i+1;

endfor

**Complexity analysis for the algorithm:**

**T(n) = Σ*1*≤*i*≤*50 (*Σ*0*≤*j*≤*49 1*) *+* Σ*0*≤*i*≤*49 = O(1)***

**A comparison between your algorithm and at least one other technique that can be used to solve the problem:**

The other technique is divide and conquer

Sample output of the solution for the different cases of the technique with proper description for the output:

**Conclusion:**