1. Member ID numbers must be between 1 and fifty thousand (50 000). When a new member is added, the program uses the following code to generate a new member ID number.

   Member ID = 1 + int(rand( )* 100000/2)

   where rand( ) returns a six-digit random number between zero and one, and int( ) returns the integer part of whatever number is in the brackets.
   If a new member is being added and rand( ) returns 0.002222 what is Member ID

2. During her analysis of the system, Jevani interviewed all the assemblers in the Kandy factory. One assembler mentioned that sometimes RoboCut would reject a good piece of timber.
   Jevani investigates this and finds that the problem started after the last software upgrade six months ago. She contacts MyCut and it claims that none of its other users have reported this problem. After some argument MyCut agrees to send Jevani the algorithms related to the software changes. She finds one algorithm that has to do with the cutting process. RoboCut uses this algorithm to check whether or not a piece of timber is long enough to use.

```
Function Check_Length(Timber_Length,
LengthRequired)
        Begin
                If Timber_Length >
        Length_Required Then
                        Return true
        Else
                        Return false
        End if
End
```

Jevani decides to test this algorithm by choosing a length of timber (Timber_Length) of 2.4 metres. For the other variable (Length_Required) she chooses the values 2.3, 2.4 and 2.5.

a. Explain why Jevani selected these values.

b. Complete the following table showing what the algorithm should return and what it actually returns.

LengthRequired	What should be returned	What is actually returned
2.3 metres		
2.4 metres		
2.5 metres		

c. Explain why RoboCut only sometimes rejects a good piece of timber.

d. State one alteration to the algorithm that would correct this error.

3. (**5 1** 4 2 8) –> (**1 5** 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1.
   (1 **5 4** 2 8) –> (1 **4 5** 2 8), Swap since 5 > 4
   (1 4 **5 2** 8) –> (1 4 **2 5** 8), Swap since 5 > 2
   (1 4 2 **5 8**) –> (1 4 2 **5 8**), Now, since these elements are already in order (8 > 5), algorithm does not swap them.

   That is one run. It should continue the same operation and sort the dataset. **Write the psudo code to this looping operation on the other side of the paper.**

   What does this sorting algorithm called?