

COMP1022Q Review Questions Week 12

Q1)

Here is a recursive function.

```
Function Recursion(ByVal InputText As String)
    If InputText = "" Then
        Recursion = 0
    Else
        Recursion = Recursion(Right(InputText, Len(InputText) - 1)) + 1
    End If
End Function
```

What will be the content of the message box if you run the following line of code?

```
MsgBox Recursion("Oh my god!")
```

Answer: The content of the message box is _____

Q2)

Here is a recursive function.

```
Function Recursion(ByVal InputText As String)
    If Len(InputText) <= 1 Then
        Recursion = InputText
    Else
        Half = Int(Len(InputText) / 2)
        Recursion = Recursion(Right(InputText, Half)) & _
                    Recursion(Left(InputText, Len(InputText) - Half))
    End If
End Function
```

What will be the content of the message box if you run the following line of code?

```
MsgBox Recursion("Oh my god!")
```

Answer: The content of the message box is _____

Q3)

You are given the following worksheet.

| | A |
|----|----|
| 1 | 11 |
| 2 | 17 |
| 3 | 14 |
| 4 | 12 |
| 5 | 3 |
| 6 | 18 |
| 7 | 14 |
| 8 | 17 |
| 9 | 4 |
| 10 | 7 |

And the following recursive function.

```
Function Recursion(ByVal First As Integer, ByVal Last As Integer)
    If First = Last Then
        Recursion = Cells(Last, 1).Value
    Else
        Dim Temp As Integer
        Temp = Recursion(First + 1, Last)
        If Cells(First, 1).Value > Temp Then
            Recursion = Cells(First, 1).Value
        Else
            Recursion = Temp
        End If
    End If
End Function
```

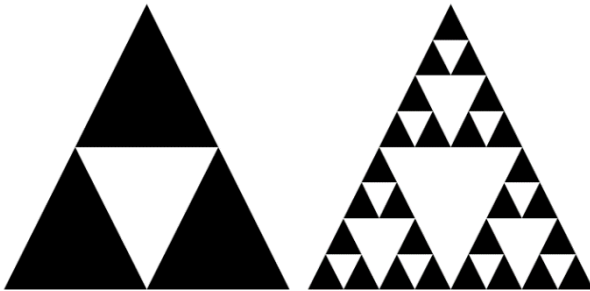
What will be the content of the message box if you run the following line of code?

```
MsgBox Recursion(1, 10)
```

Answer: The content of the message box is _____

Q4)

The picture on the left below is generated by a recursive drawing subroutine when depth is 0.



What is the depth used by the recursive drawing subroutine for the picture on the right?

- A) 1
- B) 2
- C) 3
- D) 4

Answer (A/B/C/D): _____

Q5)

Here is some VBA code.

```
Dim Numbers(10) As Integer

For Index = LBound(Numbers) To UBound(Numbers)
    Numbers(Index) = Index
Next Index

MsgBox Numbers(Int((UBound(Numbers) + LBound(Numbers)) / 2))
```

What will be the content of the message box if you run the above code?

Answer: The content of the message box is _____

Q6)

Assuming that you have created a class called **Bear**, which one of the following is a correct way to create an instance of the class?

- A) Dim GG As Bear
GG = New Bear
- B) Dim GG As Bear
Set GG = New Bear
- C) Dim GG As Bear
Set GG = New BearClass
- D) Dim GG As BearClass
GG = New BearClass

Answer (A/B/C/D): _____

Q7)

You have seen similar code like this before.

```
Dim Square As Shape
Set Square = _
    ActiveSheet.Shapes.AddShape(msoShapeRectangle, 0, 0, 100, 100)
Square.Rotation = 45
```

What is Rotation?

- A) A global variable
- B) A local variable
- C) An object's attribute
- D) An object's behaviour

Answer (A/B/C/D): _____

Q8)

Here is a VBA class called Monster.

```
Public AttackPoint As Integer
Public HitPoint As Integer

Sub Hit(ByVal AttackPoint As Integer)
    HitPoint = HitPoint - AttackPoint
End Sub

Function Dead() As Boolean
    Dead = (HitPoint <= 0)
End Function
```

Some code is then run using the class shown above.

```
Dim Pikachu As Monster
Set Pikachu = New Monster
Pikachu.HitPoint = 10
Pikachu.AttackPoint = 2

Dim Greninja As Monster
Set Greninja = New Monster
Greninja.HitPoint = 5
Greninja.AttackPoint = 3

Do While Not Pikachu.Dead() And Not Greninja.Dead()
    Pikachu.Hit Greninja.AttackPoint
    Greninja.Hit Pikachu.AttackPoint
Loop

If Pikachu.Dead() Then
    MsgBox "Oh, Pikachu is dead :~"
ElseIf Greninja.Dead() Then
    MsgBox "Yeah, Greninja is dead :)"
Else
    MsgBox "Draw!"
End If
```

If you run the above code, who will be dead according to the message shown inside the message box?

Answer (Pikachu/Greninja/None): _____

Answers to Week 12 Questions

Q1) The answer is 10.

The purpose of the recursive function is to find the length of a given string. Every time the function is called, it will recursively call itself by ‘removing’ the first letter of the given string, i.e. `Right(InputText, Len(InputText) - 1)`. Using the example, the sequence of the recursive call is then:

```
Recursion("Oh my god!")
↓
Recursion("h my god!")
↓
Recursion(" my god!")
↓
...
↓
Recursion("!")
↓
Recursion("")
```

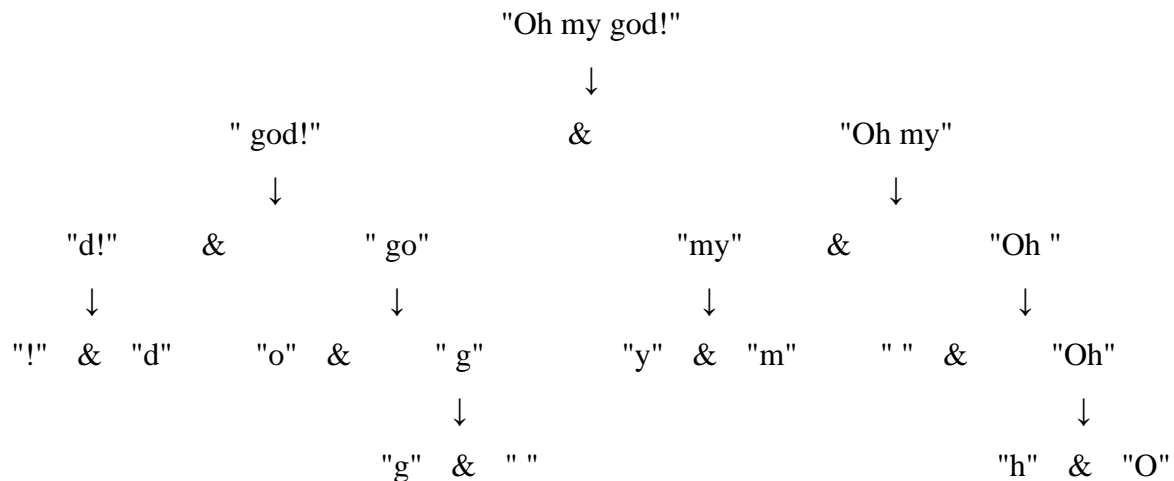
When the given string is "", the function returns 0 as "" has a length of 0. When the recursive function returns a value each time, the result is then added by 1 for the length of the first letter that has been taken out.

```
Recursion("Oh my god!")
↑ 9
Recursion("h my god!")
↑ 8
Recursion(" my god!")
↑ 7
...
↑ 1
Recursion("!")
↑ 0
Recursion("")
```

The last value returned is then 10, which is the length of “Oh my god!”.

Q2) The answer is !dog ym hO.

The purpose of the recursive function is to reverse the content of a given string. If you look into the code, you will see the recursive function subdivide the given string into two parts at the middle of the string, i.e. $\text{Int}(\text{Len}(\text{InputText}) / 2)$. Then the right half of the string is put together with the left half of the string, giving you a reverse order of these two halves. Each half of the string is then recursively applied with the same operation and therefore each half would get reversed as well. Using the example, the sequence of the recursive call is then:



If you then combine the string from the final recursion level back to the start of the call, you will be able to get the reversed version of the given string.

Q3) The answer is 18.

The purpose of the recursive function is to find the maximum value within the given rows. When the function is called recursively, the first row is not included. The number in the first row, i.e. `Cells(First, 1).Value`, is then compared against the maximum of the rest of the rows, i.e. `Recursion(First + 1, Last)`. If the number in the first row is bigger, that number is returned; otherwise, the maximum number from the rest of the rows is then returned.

Q4) The answer is B.

This can be done by counting the white triangles. At each recursive call, each black triangle is replaced by the pattern shown at depth 0. Therefore, three smaller white triangles are produced at the next depth level. Since the triangles get smaller twice, the depth is then 2.

Q5) The answer is 5.

The code fills the items inside the Numbers array with values equivalent to the index of the items. That means Numbers(0) is 0, Numbers(1) is 1 and so on. Then, the message box shows the content of the item with index (UBound(Numbers) + LBound(Numbers)) / 2. The index is simply $(10 + 0) / 2 = 5$ so the content of the message box is 5.

Q6) The answer is B.

The class name has to match and you need to use Set when you store an object in a variable.

Q7) The answer is C.

Q8) The answer is Greninja.

The loop in the code continuously 'hits' Pikachu and then 'hits' Greninja until one of them is dead. Greninja's HitPoint is smaller than or equal to 0 when it gets hit 3 times whereas the same applies to Pikachu only when it gets hit 4 times. Therefore, Greninja is dead before Pikachu.