Zadania z VBA 1

Jakub Gismatullin

Zadanie 1 Poniższy kod wypisuje w pierwszej kolumnie n pierwszych wyrazów ciągu Fibonacciego, dla podanego n.

```
Sub FibonacciNumbers()
    Dim i As Integer
    Dim Limit As Long
    Columns(1).ClearContents
    Limit = InputBox("Podaj n")
    i = 1
    Cells(i, 1).Value = 0
    i = i + 1
    Cells(i, 1).Value = 1
    Do While i < Limit
        i = i + 1
        Cells(i, 1).Value = Cells(i - 1, 1).Value + Cells(i - 2, 1).Value
    Loop
End Sub
```

(a) Zmodyfikuj ten kod aby po podaniu przez użytkownika $n \in \mathbb{N}$ oraz dowolnych liczb a, b, w pierwszej kolumnie obliczany był ciąg typu Fibonacciego, o warunkach początkowych a i b:

$$f_{n+1} = f_n + f_{n-1}, \;\; f_0 = a, \;\; f_1 = b.$$

(b) Wiadomo, że klasyczny ciąg Fibonacciego $(F_n)_{n\in\mathbb{N}}$ rośnie wykładniczo i jego tempo wzrostu jest równe

$$\lim_{n\to\infty} \sqrt[n]{F_n} = \frac{1+\sqrt{5}}{2}.$$

Oszacuj tempo wzrostu ciągu z (a) w zależności od parametrów a i b. Możesz, w tym celu, automatycznie wpisywać do drugiej kolumny kolejno liczby $\sqrt[n]{|f_n|}$.

Zadanie 2 Niech $n \in \mathbb{N}$. n-*Liczba Armstronga*[1], to taka liczba naturalna, mająca n cyfr, która jest równa sumie swoich cyfr podniesionych do n-tej potęgi. Przykładem takiej liczby jest 370, ponieważ $370 = 3^3 + 7^3 + 0^3$. Ciąg liczb Armstronga tworzą[2]:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634 \ldots.$$

Poniższy znajduje wszystkie n-liczby Armstronga, dla podanego przez użytkownika n:

```
Sub Armstrong()
    Dim kNum As Byte, Num As Long
    Dim Limit As Long
    Dim Digit As Byte, i As Byte
    Dim DigitPower As Long, Sum As Long
    Dim Counter As Byte
```

[1] https://pl.wikipedia.org/wiki/Liczby_Armstronga
[2] https://oeis.org/A005188

```
    Dim NumStr As String, Msg As String
    kNum = InputBox("Wpisz parametr n")
    Limit = (10 ^ kNum) - 1
    For Num = 1 To Limit
        NumStr = CStr(Num)
        For i = 1 To Len(NumStr)
            Digit = Mid(NumStr, i, 1)
            DigitPower = Digit ^ Len(NumStr)
            Sum = Sum + DigitPower
        Next i
        If Num = Sum And Len(CStr(Num)) = kNum Then
            Msg = Msg & Num & ", "
            Counter = Counter + 1
        End If
        Sum = 0
    Next Num
    Select Case Counter
        Case 0
            MsgBox "Nie ma " & kNum & "-cyfrowych liczb Armstronga."
        Case 1
            Msg = Left(Msg, Len(Msg) - 2)
            MsgBox "Jest jedna " & kNum _
                & "-cyfrowa liczba Armstronga: " & Msg & "."
        Case 2, 3, 4
            Msg = Left(Msg, Len(Msg) - 2)
            MsgBox "S ą " & Counter & " " & kNum _
                & "-cyfrowe liczby Armstronga: " & Msg & "."
        Case Is > 4
            Msg = Left(Msg, Len(Msg) - 2)
            MsgBox "Jest " & Counter & " " & kNum _
                & "-cyfrowych liczb Armstronga: " & Msg & "."
    End Select
End Sub
```

(a) Zmodyfikuj ten kod do znajdowania ciągu `https://oeis.org/A023052`. Tzn. aby po podaniu dwóch parametrów n i m, znajdował wszystkie n-cyfrowe liczby, które są równie sumie m-tych potęg swoich cyfr.

(b) Znajdź wszystkie liczby Münchhausena `https://en.wikipedia.org/wiki/Perfect_digit-to-digit_invariant`.