

# Laborator VBA pentru Excel 4

## Tablouri în VBA

În laboratorul precedent am văzut cum putem apela funcții simple care acceptă ca parametrii vectori, respectiv cum putem returna vectori în foaia Excel. În cazul tablourilor cu două dimensiuni, lucrurile se simplifică, nefiind necesară folosirea funcției de librărie TRANSPOSE pentru a transforma vectorul VB într-un tablou Excel.

## Funcții Simple care returnează o valoare

### Suma elementelor de deasupra diagonalei principale:

1. Lansați Microsoft Excel
2. Lansați Visual Basic Editor
3. In VBE adăugați un modul **Module1**
4. Tastați codul de mai jos in modulul **Module1**

```
Option Explicit
Function SumaDiagSup(rng As Range) As Double
    Dim Arr()
    Dim Suma As Double
    Dim R As Integer, C As Integer
    Dim i As Integer, j As Integer

    R = rng.Rows.Count
    C = rng.Columns.Count

    Arr = rng

    Suma = 0
    For i = 1 To R
        For j = i To C
            Suma = Suma + Arr(i, j)
        Next j
    Next i
    SumaDiagSup = Suma
End Function
```

5. Lansați comanda **Compile VBAProject** din meniul **Debug** pentru a verifica codul
6. Activați foaia de calcul Excel și inserați următoarea serie de valori:

	A	B	C	D
1	1	2	3	
2	4	5	6	

3	7	8	9	
4	10	11	12	

- Activați foaia de calcul Excel și inserați formula **SumaDiagSup** într-o celulă goală din foaia de calcul Excel, dându-i ca parametru plaja de celule **A1:C4**.

**Obs** Remarcați că am preluat elementele plajei de celule (Range) **rng** printr-o singură instrucțiune de atribuire în linia **Arrr = rng** pentru că **Arr()** a fost definit ca **Variant**.

Observăm că algoritmul **SumaDiagSup** nu are nici o problemă dacă matricea nu este pătratică.

- Modificați funcția **SumaDiagSup** astfel încât în cazul unei matrice nepătratice, funcția să returneze un mesaj de eroare.

### Maximul elementelor de deasupra diagonalei principale:

- Creați o nouă funcție denumită **MaxDiagSup** în modulul **Module1**.
- Adăugați codul care ne va returna elementul maxim de deasupra diagonalei principale, după modelul funcției **SumaDiagSup**.
- Activați foaia de calcul Excel și inserați formula **MaxDiagSup** într-o celulă goală din foaia de calcul Excel, dându-i ca parametru plaja de celule **A1:C4**.
- Modificați funcția **MaxDiagSup** astfel încât în cazul unei matrice nepătratice, funcția să returneze un mesaj de eroare

```
Function MaxDiagSup(rng As Range) As Variant
    Dim Arr()
    Dim Max As Double
    Dim R As Integer, C As Integer
    Dim i As Integer, j As Integer

    R = rng.Rows.Count
    C = rng.Columns.Count

    Arr = rng

    Max = Arr(1, 1)
    For i = 1 To R
        For j = i To C
            If Arr(i, j) > Max Then
                Max = Arr(i, j)
            End If
        Next j
    Next i
    MaxDiagSup = Max
End Function
```

## Funcții Simple care returnează un tablou

### Înmulțirea cu un scalar

1. Adăugați funcția alăturată în modulul **Module1**.
2. Activați foaia de calcul Excel și inserați formula **ScaleArr** într-o celulă liberă, selectând aceeași plajă de celule ca în exemplul precedent și dați valoarea 3 scalarului sc =**ScaleArr(A1:C4,3)**.
3. Selectați un grup de 9 celule (3x3) cu celula care conține formula noastră în colțul stânga sus.
4. Transformați selecția într-o formulă matrice.

```
Function ScaleArr(rng As Range, ByVal sc As Double) As Variant
    Dim Arr()
    Dim R As Integer, C As Integer
    Dim i As Integer, j As Integer

    R = rng.Rows.Count
    C = rng.Columns.Count

    Arr = rng

    For i = 1 To R
        For j = 1 To C
            Arr(i, j) = Arr(i, j) * sc
        Next j
    Next i
    ScaleArr = Arr
End Function
```

**Obs:** Remarcăm că atât tabloul **Arr()** cât și funcția **ScaleArr** au fost definite ca **Variant**.

Dacă declarăm tabloul **Arr()** ca **Variant** putem face operația de atribuire **Arr = rng**. Dacă tabloul **Arr()** nu este declarat ca **Variant**, la atribuirea **Arr = rng**, se va produce o eroare.

### Produsul a 2 matrice

Dacă matricele care se înmulțesc sunt A(m,n) și B(n,p), matricea rezultantă C=A·B va fi de rang (m,p).

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{bmatrix} \cdot \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,p} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1,1} & b_{n-1,2} & \cdots & b_{n-1,p} \\ b_{n,1} & b_{n,2} & \cdots & b_{n,p} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,p} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,p} \end{bmatrix}$$

Un element  $c_{i,j}$  al matricei C va fi obținut prin relația:

$$c_{i,j} = a_{i,1} \cdot b_{1,j} + a_{i,2} \cdot b_{2,j} + \cdots + a_{i,n-1} \cdot b_{n-1,j} + a_{i,n} \cdot b_{n,j} = \sum_{k=1}^n a_{i,k} \cdot b_{k,j}$$

## 1. Adăugați funcția **ProdusMatrice** în modulul **Module1**.

```
Function ProdusMatrice(rng1 As Range, rng2 As Range) As Variant
    Dim Arr1(), Arr2(), Arr3() As Double
    Dim row1 As Integer, col1 As Integer
    Dim row2 As Integer, col2 As Integer
    Dim i As Integer, j As Integer, k As Integer

    row1 = rng1.Rows.Count
    col1 = rng1.Columns.Count
    row2 = rng2.Rows.Count
    col2 = rng2.Columns.Count

    If col1 <> row2 Then
        ProdusMatrice = "Invalid"
        Exit Function
    End If

    ReDim Arr3(1 To row1, 1 To col2)

    Arr1 = rng1      Arr2 = rng2

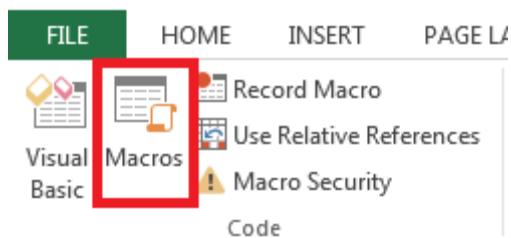
    For i = 1 To row1
        For j = 1 To col2
            Arr3(i, j) = 0
            For k = 1 To col1
                Arr3(i, j) = Arr3(i, j) + Arr1(i, k) * Arr2(k, j)
            Next k
        Next j
    Next i

    ProdusMatrice = Arr3
End Function
```

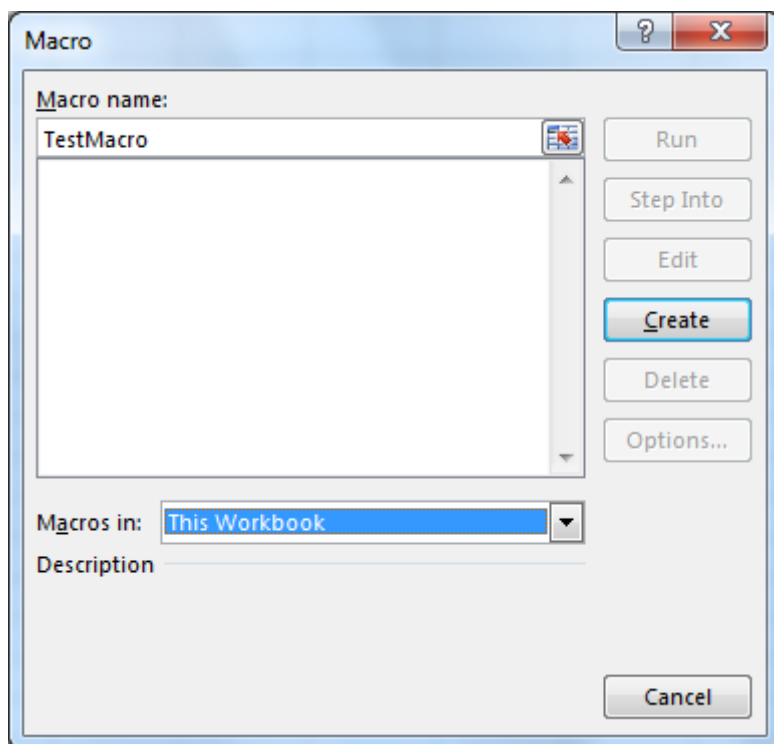
2. Activați foaia de calcul Excel și creați două tablouri care să respecte condiția ca numărul de linii al primului tablou să fie egal cu numărul de coloane al celui de-al doilea.
3. Inserați formula simplă **ProdusMatrice** într-o celulă liberă a foii.
4. Selectați un grup de celule de forma  $m \times p$  (numărul de rânduri al primei matrice  $\times$  numărul de coloane al celei de-a doua matrice) astfel încât celula care conține formula **ProdusMatrice** să se găsească în colțul din stânga sus al selecției.
5. Transformați formula într-o formulă matrice în toată selecția.

## Automatizarea Excel

1. Lansați comanda **Macros** din fila **DEVELOPER**.



2. În fereastra **Macro**, tastați **TestMacro** în caseta **Macro name**; selectați opțiunea **This Workbook** din lista derulantă **Macros in**: și clic **Create**.



3. Adăugați codul alăturat în subrutina **TestMacro** și lansați comanda **Compile VBAProject** din meniul **Debug** pentru a corecta erorile.
4. Aranjați ferestrele Excel și VBA astfel încât să fie ambele vizibile pe ecran.
5. Adăugați un punct de oprire în linia **Texte(1, 1)...** și executați codul pas cu pas cu combinația de taste **SHIFT+F8**.

```
Sub TestMacro()  
    Dim wks As Worksheet  
    Dim Texte(1 To 3, 1 To 1) As String  
    Dim Valori(1 To 3, 1 To 1) As Integer  
  
    Texte(1, 1) = "Apa"  
    Texte(2, 1) = "Incalzire"  
    Texte(3, 1) = "Electricitate"  
    Valori(1, 1) = 125  
    Valori(2, 1) = 350  
    Valori(3, 1) = 75  
  
    Set wks = Application.ActiveWorkbook.Sheets.Add  
    wks.Range("A1:A3").Value = Texte  
    wks.Range("B1:B3").Value = Valori
```

```
wks.Cells(4, 1).Value = "TOTAL"  
wks.Cells(4, 1).Font.Bold = True  
wks.Range("B4").Formula = "=SUM(B1:B3) "
```

```
End Sub
```

## Bonus

1. Încărcați documentul Excel tetris.xls și Apăsați butonul Play Game.