VBA in Excel 2

Vectors in VBA

Functions that return a single value. The sum of a range.

To work with vectors in VBA we'll pass as parameter of a function an object of type **Range** (selection).

1. Start Microsoft Excel 2013
2. Start Visual Basic Editor
3. In VBE Insert a new module Module1
4. Add the following code to the module window.

```
Option Explicit

Function SumVector(rng As Range) As Double
    Dim i As Integer
    Dim n As Integer
    Dim Sum As Double

    n = rng.Cells.Count
    ReDim Vect(1 To n) As Double

    For i = 1 To n
        Vect(i) = rng(i)
        Sum = Sum + Vect(i)
    Next i
    SumVector = Sum
End Function
```

5. Launch **Compile VBAProject** from the **Debug** menu to check for errors.
6. Activate the Excel sheet and insert the following values:

	A	B	C	D
1	1			
2	3			
3	5			
4	7			
5	9			
6	11			

7. Move the cursor to cell B6 and insert the newly created function **SumVector** from the dropdown list with available functions from the group **User Defined**.

8. In the **Function Arguments** dialog, select the range A1:A6 as input and click OK

Remarks

Array variables can be static or dynamic. That's determined by the **Dim** statement. If you specify dimensions when you declare the variable, it's static and always will be. If you leave the dimensions blank in the Dim statement, it's dynamic and can be changed.

```
Dim Array1(1 To 10) As String 'static array
Dim Array2() As String 'dynamic array
```

Dynamic arrays can be changed using the **Redim** statement.

```
Dim Arr1() As Double
ReDim Arr1(Selection.Columns.Count, Selection.Rows.Count)
```

Functions that return a single value. The max of a vector.

1. Add the following code to the Module1 window.

```
Public Function MaxVector(rng As Range) As Double
    Dim i As Integer
    Dim n As Integer
    Dim Max As Double

    n = rng.Cells.Count
    ReDim Vect(1 To n) As Double
    For i = 1 To n
        Vect(i) = rng(i)
    Next i

    Max = Vect(1)
    For i = 2 To n
        If (Vect(i) > Max) Then
            Max = Vect(i)
        End If
    Next i

    MaxVector = Max
End Function
```

2. Activate the Excel sheet and insert the function **MaxVector** with the same range as parameter as in the previous example (A1:A6) in cell **C6**

3. Add a breakpoint in the beginning of the function (the first executable statement) and execute step by step the code, adding the variables n, I, Vect and Max to the Watch window.

Functions that return a vector. Scaling a vector.

1. Add the following code to the Module1 window.

```
Function ScaleVector(rng As Range, sc As Double) As Double()
    Dim i As Integer
    Dim n As Integer

    n = rng.Cells.Count
    ReDim Vect(1 To n) As Double
    For i = 1 To n
        Vect(i) = rng(i)
    Next i

    ReDim Scaled(1 To n) As Double
    For i = 1 To n
        Scaled(i) = sc * Vect(i)
    Next i

    ScaleVector = Scaled
End Function
```

2. Activate the Excel sheet and insert the function ScaleVector in cell D1 using the same range A1:A6 for the first parameter and 3 for the scalar (=ScaleVector(A1:A6,3)).

Remarks

You'll notice that the ScaleVector function inserts only the first value from the returned vector. To insert the rest of the vector's items we need to use an array formula.

To insert an array formula:

3. Select the range of cells **D1:I1** starting from cell **D1** that contains the inserted formula.
4. Edit the cell **D1** by pressing **F2**.
5. Finish the editing by pressing the combination **CTRL+SHIFT+ENTER**.

To insert a vector vertically:

6. Insert the function **ScaleVector** in cell **D2** with the same data range (A1:A6).
7. Modify the inserted function by wrapping it in the function TRANSPOSE. The formula will become **=TRANSPOSE(ScaleVector(A1:A6,3))**
8. Select the range of cells **D2:D7** starting from cell **D2** that contains the inserted formula.

9. Edit the cell **D2** by pressing **F2**.
10. Finish the editing by pressing the combination **CTRL+SHIFT+ENTER**.

The sum of two vectors.

1. Add the following code to the Module1 window.

```
Function Sum2Vec(rng1 As Range, rng2 As Range) As Variant
    Dim i As Integer
    Dim n1 As Integer
    Dim n2 As Integer

    n1 = rng1.Cells.Count
    n2 = rng2.Cells.Count
    If (n1 <> n2) Then
        Sum2Vec = "Different sizes."
        Exit Function
    End If
    ReDim Vect(1 To n1) As Double

    For i = 1 To n1
        Vect(i) = rng1(i) + rng2(i)
    Next i
    Sum2Vec = Vect
End Function
```

2. Activate the Excel sheet and insert the function Sum2Vec giving two ranges as input parameters.
3. Use the array formula to insert the sum of the two ranges as presented in the previous example.

Sorting a vector. Bubble sort.

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Example:
First Pass:
(**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.

Second Pass:
(**1 4** 2 5 8) –> (**1 4** 2 5 8)
(1 **4 2** 5 8) –> (1 **2 4** 5 8), Swap since 4 > 2

(1 2 **4 5** 8) –> (1 2 **4 5** 8)
(1 2 4 **5 8**) –> (1 2 4 **5 8**)

1. Add the following code to the Module1 window.

```
Public Function Bubble(rng As Range)
    Dim temp As Double
    Dim i As Integer, j As Integer
    Dim n As Integer

    n = rng.Cells.Count
    ReDim A(1 To n)
    For i = 1 To n
        A(i) = rng(i)
    Next i

    For i = 1 To n - 1
        For j = 1 To n - i
            If (A(j) > A(j + 1)) Then
                temp = A(j)
                A(j) = A(j + 1)
                A(j + 1) = temp
            End If
        Next j
    Next i
    Bubble = A
End Function
```

2. Activate the Excel sheet and add the series 1, 5, 4, -7, 2, -3 in the cells **J1:J5**.
3. Insert the function Bubble in cell **K1** with the range J1:J5 (**=Bubble(J1:J5)**).
4. Add a breakpoint in the line `For i = 1 To n - 1` and add the variables temp and the vector A to the Watch window.
5. Execute the function step by step and inspect the values of the vector after each outer loop (about i).
6. Use the transpose technique to insert all the values returned by the Bubble function in the Excel sheet.