

Visual Basic for Applications Programming

Damiano SOMENZI

School of Economics and Management

Advanced Computer Skills

`damiano.somenzi@unibz.it`

Week 10



Outline

1 Algorithms

- Sorting
- Exercises: exchange values, selection and sorting

Outline

1 Algorithms

- **Sorting**

- Exercises: exchange values, selection and sorting

Algorithms

Sorting

Sorting

Sorting is a classical and important algorithmic problem

for any given input the algorithm halts with the output that is a permutation of $a_1, a_2, a_3, \dots, a_n$: $b_1, b_2, b_3, \dots, b_n \Rightarrow b_1 < b_2 < b_3 < \dots < b_n$

INPUT: sequence of n numbers $a_1, a_2, a_3, \dots, a_n$ (5, 2, 4, 10, 7)

OUTPUT: a permutation of the input sequence of numbers $b_1, b_2, b_3, \dots, b_n$ (2, 4, 5, 7, 10)

selection sort: strategy

```
Dim i_min As Long, i As Long
For i = LBound(item) To UBound(item) - 1 Step 1
    i_min = minimum(item, i, UBound(item))
    Call exchange (item(i), item(i_min))
next i
```

- **item:** the array of elements to order
- **minimum:** the function returns an index between i and UBound(item)

bubble sort: strategy

```
Dim i As Long, j As Long
For i = LBound(item) To UBound(item) - 1
    For j = UBound(item) To i + 1 Step -1
        If item(j) < item(j - 1) Then
            Call exchange (item(j), item(j - 1))
        End if
    next j
next i
```

- **item:** the array of elements to order

Outline

1 Algorithms

- Sorting

- Exercises: exchange values, selection and sorting

Exercise

one

An investigation returned color (hair and eyes) associations of 50 subjects as reported in this picture

	brown	blue	green	hazel
black	9	0	5	1
brown	7	3	1	1
blond	5	8	8	0
red	0	2	0	0

A VBA tool should be able to transpose the rows (hair) and the columns (eyes), reporting data as in the next picture

	black	brown	blond	red
brown	9	7	5	0
blue	0	3	8	2
green	5	1	8	0
hazel	1	1	0	0

hint: it could be helpful a procedure that exchanges values of two cells

Exercises

```
Sub exLab(c1 As Range, c2 As Range)
    'given two cells as argument
    'the subroutine copies the value (String)
    'of the first cell into the second and vice versa
    Dim temp As String
    temp = c1.Value
    c1.Value = c2.Value
    c2.Value = temp
End Sub

Sub exNum(c1 As Range, c2 As Range)
    'given two cells as argument
    'the subroutine copies the value (Long)
    'of the first cell into the second and vice versa
    Dim temp As Long
    temp = c1.Value
    c1.Value = c2.Value
    c2.Value = temp
End Sub

Sub transpose()
    'given a group of cells (rows = columns)
    'the subroutine transposes rows and columns
    Dim r As Long, c As Long
    For r = 1 To 5 'for row and columns names
        Call exLab(Cells(r, 1), Cells(1, r))
    Next r
    For r = 2 To 5 'for numbers
        For c = r To 5
            Call exNum(Cells(r, c), Cells(c, r))
        Next c
    Next r
End Sub
```

Exercise

two

Every month we register the scheduled meetings in a single worksheet. For example the picture below shows the meetings scheduled in January. Of course we should be able to register meetings as soon as they are scheduled. The worksheet is named with the month name.

	A	B	C	D	E
1	meetings				
2	date	start	end	place	notes
3	05.01.12	10:00	12:30	Room A10	Formal
4	05.01.12	15:00	16:00	Room C02	
5	12.01.12	10:00	11:00	Room A10	Formal
6	13.01.12	14:00	15:00	Room C02	
7	19.01.12	15:00	16:00	Room C02	
8	19.01.12	17:00	18:00	Room B09	Kick Off
9	23.01.12	9:30	10:30	Room B09	
10	23.01.12	15:30	17:30	Room B09	
11	25.01.12	12:00	13:00	Room A10	
--					

We need a VBA tool in order to display in a single message box the next scheduled meetings for the month specified by the user. For example if we run the tool today (12/01/2012) the message should display

```
13.01.12 14:00 15:00 Room C02
19.01.12 15:00 16:00 Room C02
19.01.12 17:00 18:00 Room B09
23.01.12 9:30 10:30 Room B09
23.01.12 15:30 17:30 Room B09
25.01.12 12:00 13:00 Room A10
```

Exercises

1/3

```
Function nextM(ByVal d As Date) As Boolean
    'the function returns TRUE whether the date meeting will be in the future
    'FALSE, otherwise
    If d < Date Then
        nextM = False
    Else
        nextM = True
    End If
End Function
```

Exercises

2/3

```
Sub displayAgenda(agenda As Worksheet)
    'the subroutine displays date, start, end and place registered in the worksheet
    'but only if the function "nextM" returns TRUE
    'start and end are registered as string
    Dim message As String
    Dim i As Integer
    message = ""
    i = 3
    Do While agenda.Cells(i, 1).Value > 0
        If nextM(agenda.Cells(i, 1).Value) = True Then
            message = message & agenda.Cells(i, 1).Value & " " & _
                TimeValue(agenda.Cells(i, 2).Value) & " " & _
                TimeValue(agenda.Cells(i, 3).Value) & " " & _
                agenda.Cells(i, 4).Value & Chr(13)

            End If
            i = i + 1
        Loop
        If message = "" Then
            MsgBox ("no more meetings this month")
        Else
            MsgBox (message)
        End If
    End Sub
```

Exercises

3/3

```
Sub nextMeetings()  
    'the subroutine asks for the name of the month of interest  
    'then it searches for this month among the worksheet names  
    Dim month As String  
    Dim i, ws As Integer  
    month = InputBox("Please, for which month?")  
    ws = 0  
    For i = 1 To Worksheets.Count  
        If UCase(Worksheets(i).Name) = UCase(month) Then  
            ws = i  
            Exit For  
        End If  
    Next i  
    If ws = 0 Then  
        MsgBox ("Your month does not exist in your agenda")  
    Else  
        Call displayAgenda(Worksheets(ws))  
    End If  
End Sub
```

Exercise

three

Monthly temperatures of four places are registered into a spreadsheet. A single worksheet collects temperatures of one month, for example the sheet 1 collects temperatures of January, as shown in the figure below. The range object Range ("C3:F33") could refer to the set of temperatures.

	A	B	C	D	E	F	
1	place						
2	month	day	Bozen	Innichen	Sterzing	Mals	
3	1	1	-3	-7	-6	-2	
4	1	2	-3	-7	-6	-2	
5	1	3	-3	-9	-9	-4	
6	1	4	-1	-7	-6	-2	
7	1	5	-3	-7	-6	-2	
8	1	6	-2	-9	-9	-4	
9	1	7	0	-3	-4	-1	
10	1	8	-5	-4	-3	0	

We need a tool in order to perform the following tasks:

- 1 For each place collect the temperatures of the month
- 2 Compute the average temperature of the month
- 3 Display place names and the corresponding average temperatures (increasing order)
- 4 It could happen that a temperature is unavailable, therefore that day should not be used for the average computation

Exercises

1/3

```
Function avTemp(temp() As Single)
    'given the temperatures of the month held in an array of numbers
    'the function returns the corresponding average value
    'temperatures marked with the number "100" are not considered
    Dim i As Integer
    Dim d As Integer
    Dim t As Single
    d = 0
    t = 0
    For i = LBound(temp) To UBound(temp)
        If temp(i) <> 100 Then
            d = d + 1
            t = t + temp(i)
        End If
    Next i
    avTemp = Round(t / d, 2)
End Function
```

Exercises

2/3

```
Sub weather()  
    'the subroutine analyses the temperatures held in the active worksheet  
    'firstly it computes for each place its average  
    'then it displays the averages in increasing order  
    Dim h, r, i, j, t, p As Integer  
    Dim message As String  
    message = ""  
    Dim table As Range  
    Set table = Worksheets(1).Range("C3:F33")  
    Dim temp(1 To 31) As Single 'temperatures of one month: max 31 days  
    Dim av(1 To 4) As Single 'average place: max 4 places  
    Dim place(1 To 4) As Integer 'column place: max 4 places  
    For h = 1 To 4 Step 1 'for each place  
        For r = 1 To 31 Step 1 'for each temperature  
            If table.Cells(r, h).Value <> "" Then  
                temp(r) = table.Cells(r, h).Value  
            Else  
                temp(r) = 100  
            End If  
        Next r  
        av(h) = avTemp(temp) 'average -> the array  
    Next h  
    For i = 1 To 4  
        place(i) = i 'set array of places  
    Next i  
  
    ... cont.
```

Exercises

3/3

```
... cont.  
  
'sorting: bubble sort strategy  
For i = 1 To 3 Step 1  
  For j = 4 To i + 1 Step -1  
    If av(j) < av(j - 1) Then  
      'exchange the average  
      t = av(j)  
      av(j) = av(j - 1)  
      av(j - 1) = t  
      'exchange the place  
      p = place(j)  
      place(j) = place(j - 1)  
      place(j - 1) = p  
    End If  
  Next j  
Next i  
For h = 1 To 4 Step 1  
  message = message & "Place: " & place(h) & " - " & "Average: " & av(h) & Chr(13)  
Next h  
MsgBox (message)  
End Sub
```