Visual Basic for Applications

Programming

Damiano SOMENZI

School of Economics and Management

Advanced Computer Skills

damiano.somenzi@unibz.it

Week 11

Outline

1. Algorithms
   - Exercises: Searching and Sorting

Outline

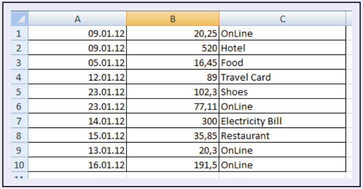
Exercises

one

In order to be able to check quickly the credit card expenses, each payment should be registered into a worksheet, for example as in the picture below a single worksheet collects the payments of one month.

For each entry are reported the date, the amount and a note.

We could assume that this data set has no more than 10 entries.



	A	B	C
1	09.01.12	20,25	OnLine
2	09.01.12	520	Hotel
3	05.01.12	16,45	Food
4	12.01.12	89	Travel Card
5	23.01.12	102,3	Shoes
6	23.01.12	77,11	OnLine
7	14.01.12	300	Electricity Bill
8	15.01.12	35,85	Restaurant
9	13.01.12	20,3	OnLine
10	16.01.12	191,5	OnLine

A tool should be provided in order to perform the following tasks:

1. Sort the payments by date
2. Copy just the *OnLine* payments into a new worksheet

Exercises

1/3

```
Function minItem(ByVal col As Long, ByVal st As Long, ByVal ed As Long) As Long
    ’given a column of the worksheet, starting row and ending row
    ’the function returns the row where the minimum value is held
    ’this function supports the selction sort strategy
    Dim min As Long
    Dim i As Long
    min = st
    For i = st + 1 To ed Step 1
        If Cells(i, col).Value < Cells(min, col).Value Then
            min = i
        End If
    Next i
    minItem = min
End Function
```

Exercises

2/3

```
Sub exchangeDate(h As Range, k As Range)
    ’the subroutine moves the value of the first cell
    ’into the second cell and vice versa
    Dim temp As Date
    temp = h.Cells(1, 1).Value
    h.Cells(1, 1).Value = k.Cells(1, 1).Value
    k.Cells(1, 1).Value = temp
End Sub

Sub exchangeAmount(h As Range, k As Range)
    ’the subroutine moves the value of the first cell
    ’into the second cell and vice versa
    Dim temp As Double
    temp = h.Cells(1, 1).Value
    h.Cells(1, 1).Value = k.Cells(1, 1).Value
    k.Cells(1, 1).Value = temp
End Sub

Sub exchangeNote(h As Range, k As Range)
    ’the subroutine moves the value of the first cell
    ’into the second cell and vice versa
    Dim temp As String
    temp = h.Cells(1, 1).Value
    h.Cells(1, 1).Value = k.Cells(1, 1).Value
    k.Cells(1, 1).Value = temp
End Sub
```

Exercises

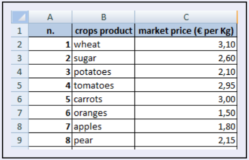
3/3

```
Sub sorting()
    'data held in the data set are sorted by date (increasing order)
    'it applies the selection sort strategy
    Dim imin As Long, i As Long
    For i = 1 To 9 Step 1
       imin = minItem(1, i, 10)
       Call exchangeDate(Cells(i, 1), Cells(imin, 1))
       Call exchangeAmount(Cells(i, 2), Cells(imin, 2))
       Call exchangeNote(Cells(i, 3), Cells(imin, 3))
    Next i
End Sub
Sub selPay()
    'it copies data that refers to OnLine payments
    'into a new worksheet
    Dim r As Long, i As Long
    i = 1 'set the first row of the new worksheet
    For r = 1 To 10
        If UCase(Cells(r, 3).Value) = UCase("OnLine") Then
            Worksheets(2).Cells(i, 1).Value = Worksheets(1).Cells(r, 1).Value
            Worksheets(2).Cells(i, 2).Value = Worksheets(1).Cells(r, 2).Value
            Worksheets(2).Cells(i, 3).Value = Worksheets(1).Cells(r, 3).Value
            i = i + 1 'next row available in the new worksheet
        End If
    Next r
End Sub
Sub main()
    Call sorting
    Call selPay
End Sub
```

Exercises

two

A basket of few European crops, with the related market price was registered into a spreadsheet in order to know the cheapest price of a product chosen by the user. The baskets (for example Italian and French markets) could be registered into distinct worksheets. In the picture below is reported the basket of the Italian market. If a product is not into the basket we assume that the related price is equal to 0. The basket data set has no more than 15 entries, therefore we could refer the range object `Range("A2:B16")`.



	A	B	C
1	n.	crops product	market price (€ per Kg)
2	1	wheat	3,10
3	2	sugar	2,60
4	3	potatoes	2,10
5	4	tomatoes	2,95
6	5	carrots	3,00
7	6	oranges	1,50
8	7	apples	1,80
9	8	pear	2,15

A tool should be provided in order to perform the following tasks:

1. Ask for a product name
2. Select the price of this product in each market
3. Determine and display the cheapest price

Exercises

1/2

```
Function cheapest(ByVal p1 As Double, p2 As Double) As Double
    ’the function returns the cheapest price
    If p1 = 0 Then
        cheapest = p2
    ElseIf p2 = 0 Then
        cheapest = p1
    ElseIf p1 < p2 Then
        cheapest = p1
    Else
        cheapest = p2
    End If
End Function

Function itemPrice(items() As String, item As String) As Long
    ’the function returns the row where the item is held
    ’0 otherwise
    Dim i As Long
    For i = LBound(items) To UBound(items) Step 1
        If items(i) = item Then
            itemPrice = i
            Exit Function
        End If
    Next i
    itemPrice = 0
End Function
```

Exercises

2/2

```
Sub comparePrice()
    ’the subroutine asks for a product name
    ’then it selects the prices of the product in both markets
    ’finally it displays the cheapest price
    Dim prod As String
    prod = InputBox("Tell me, your product name")
    Dim w As Integer, k As Integer
    Dim r As Long
        Dim market(2 To 16) As String
    Dim best(1 To 2) As Double
    For w = 1 To 2 Step 1
        For r = 2 To 16 Step 1
            market(r) = Worksheets(w).Cells(r, 2).Value
        Next r
        k = itemPrice(market, prod)
        If k = 0 Then
            best(w) = 0
        Else
            best(w) = Worksheets(w).Cells(k, 3).Value
        End If
    Next w
    MsgBox ("product: " & prod & " - cheapest price: " & cheapest(best(1), best(2)))
End Sub
```