

Caderno 4, Ficha 2 – Offset Exercício extra

Enunciado:

Escreva uma macro que troca o conteúdo da célula activa com o valor da célula à sua direita.

Exº:

- Antes de correr a macro:

	A	B	C
1			
2		Penélope	Cruz
3			

- Depois de correr a macro:

	A	B	C
1			
2		Cruz	Penélope
3			

Resolução:

- Instrução para passar o conteúdo da célula activa para a célula à sua direita:

```
ActiveCell.Offset (0, 1) = ActiveCell
```

- Instrução para passar o conteúdo da célula à direita para a célula activa:

```
ActiveCell = ActiveCell.Offset (0, 1)
```

- No entanto, depois da primeira instrução ser executada, perde-se o conteúdo da célula activa e vice-versa:

	A	B	C
1			
2		Cruz	Cruz
3			

- Inferimos que: a primeira cópia de valores, precisamos de salvaguardar algures o conteúdo da célula activa.

(continua)

Caderno 4, Ficha 2 – Offset - Exercício extra

Resolução (Cont.):

- Se quiséssemos fazer esta operação manualmente, usaríamos uma célula auxiliar (por exemplo, B1):

	A	B	C
1		Penélope	
2		Penélope	Cruz
3			

	A	B	C
1		Penélope	
2		Cruz	Cruz
3			

	A	B	C
1		Penélope	
2		Cruz	Penélope
3			

- Podemos implementar esta estratégia em VB. No entanto, coloca-se o problema de não sabermos se a célula auxiliar (neste caso, B1) possui algum conteúdo – o qual se perderia.

O único efeito que se pretende para a macro é que troque os conteúdos das células.

- Solução: em vez de guardarmos numa célula, guardamos numa variável em VB.

`Dim x As Variant`

' Como não sabemos de que tipo (texto, número, etc.) é o conteúdo da célula, declaramos a variável como **Variant**

`x = ActiveCell`



	A	B	C
1		Penélope	
2		Penélope	Cruz
3			

`ActiveCell = ActiveCell.Offset (0, 1)`

	A	B	C
1		Penélope	
2		Cruz	Cruz
3			

`ActiveCell.Offset (0, 1) = x`



	A	B	C
1		Penélope	
2		Cruz	Penélope
3			

(cont.)

Caderno 4, Ficha 2 – Offset - Exercício extra

Resolução (Cont.): Conclusões

- Podemos pensar nas variáveis como se fossem células, nas quais podemos guardar valores temporariamente;
- As variáveis desaparecem assim que termina a macro a que pertencem

O que neste caso até se revela ainda mais vantajoso relativamente a usar uma célula para armazenamento temporário: a célula teríamos que a apagar; a variável não.

Solução:

```
Sub Trocar_celulas ()
  Dim x As Variant

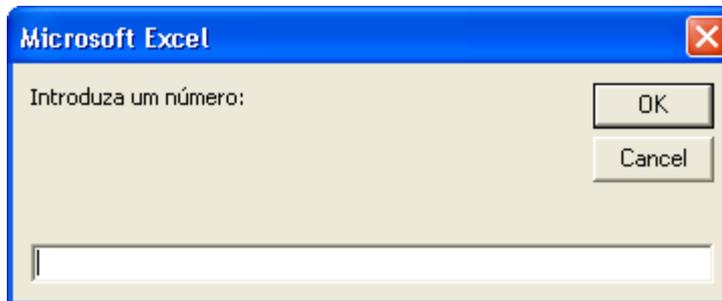
  x = ActiveCell
  ActiveCell = ActiveCell.Offset (0, 1)
  ActiveCell.Offset (0, 1) = x
End Sub
```

Caderno 4, Ficha 3 – Procura - Exercício 1

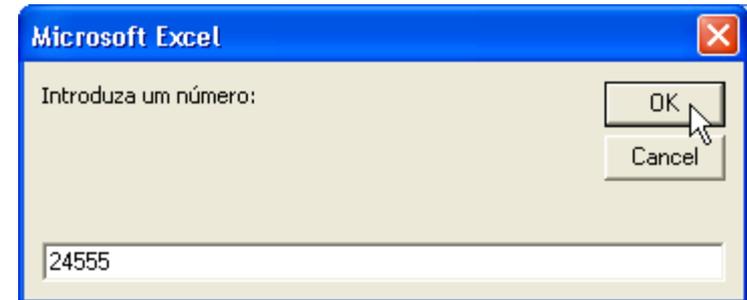
Enunciado:

Escreva uma macro que solicita um número ao utilizador e depois posiciona o Excel automaticamente na primeira célula da coluna A, a partir da célula A2, que contenha o dito número.

- Macro solicita um número:



- Utilizador escreve número e carrega em *Ok*:



- A célula activa posiciona-se na célula da coluna A onde se encontra esse número:

	A	B
1	Nº	
2	24375	
3	24349	
4	24876	
5	24769	
6	24555	
7	24696	
8	24937	

Caderno 4, Ficha 3 – Procura - Exercício 1

Resolução:

- Como faríamos nós para nos posicionarmos “a olho” na célula que possuísse o número procurado?
 1. Olhávamos para a célula A2
 2. Se o valor nessa célula não fosse igual ao número que procurávamos, olhávamos para a célula imediatamente por baixo
 3. Repetíamos o passo anterior até que a célula para onde estávamos a olhar possuísse o dito número
 4. Então activaríamos essa célula onde tínhamos parado
- Ora, vamos dar instruções em VB para o Excel fazer exactamente como nós;

Em VB:

- Precisamos de uma variável para memorizar em cada momento a célula para a qual se está a “olhar”:

```
Dim c As Range
```

` Ao indicarmos que a variável é do tipo Range, indicamos que serve para guardar valores que representam células

c

- Indicamos que a primeira célula a inspeccionar é a célula A2:

```
Set c = Range ("A2")
```

c

	A	B
1	Nº	
2	24375	
3	24349	
4	24876	
5	24769	

Caderno 4, Ficha 3 – Procura - Exercício 1

Continuação:

- Se a célula indicada por **c** não possuir o número que procuramos, i.é, se:

c.Value <> num

- ...indicamos que a nova célula a inspeccionar é a que está imediatamente a baixo – a variável **c** deverá passar a referir a célula que está por baixo:

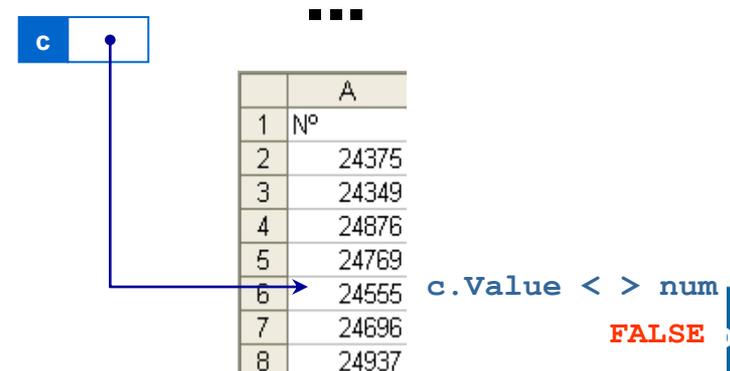
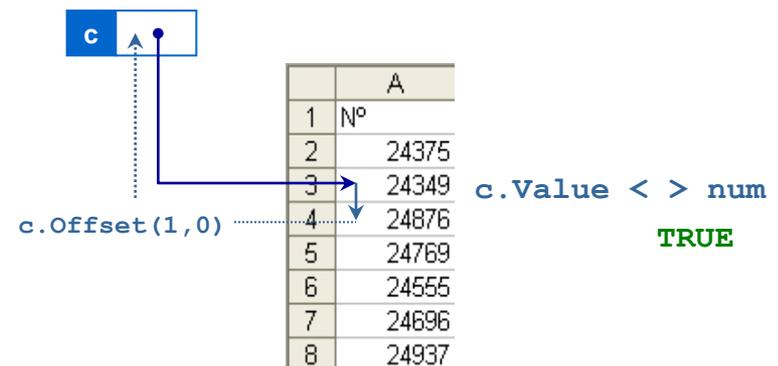
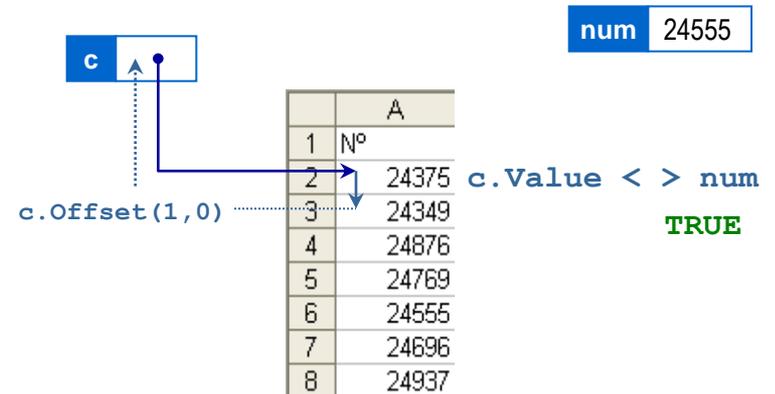
Set c = c.Offset (1, 0)

- Para repetir a instrução anterior enquanto **c** não estiver a referir uma célula com o número procurado:

Do While c.Value <> num

Set c = c.Offset (1, 0)

Loop



Exercício Extra 1 – Parte 1

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

- Fazer uma macro que preenche a célula G3 com “Passou” ou com “Chumbou”
 - O/a aluno/a só terá passado se tiver nota de avaliação contínua igual ou superior a 10...
 - ...e simultaneamente se a nota da frequência for igual ou superior a 7,5 (a nota mínima)
 - Não considere ainda a existência de nota de exame.

Exercício Extra 1 – Parte 1

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

```

Sub Passou_chumbou ()
  IF Range ("E3") >= 10 AND Range ("D3") >= 7.5 THEN
    Range ("G3") = "Passou"
  ELSE
    Range ("G3") = "Chumbou"
  END IF
End Sub
    
```

Exercício Extra 1 – Parte 2

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

- Considere agora também a nota de exame.
 - Mesmo que o/a aluno/a não tenha obtido aprovação na avaliação contínua, se tiver nota de exame igual ou superior a 9,5 obtém aprovação

Exercício Extra 1 – Parte 2

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

```

Sub Passou_chumbou ()
  IF Range("E1") >= 10 AND Range("D1") >= 7.5 OR Range("F1") >= 10 THEN
    Range("G1") = "Passou"
  ELSE
    Range("G1") = "Chumbou"
  END IF
End Sub
    
```

Exercício Extra 1 – Parte 3

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

- Transformar a macro anterior numa função
 - Comece por definir que parâmetros serão necessários para esta função
- Usar essa função em toda a coluna G, referente a outros alunos

Exercício Extra 1 – Parte 3

	A	B	C	D	E	F	G
1	Nome	Mini-teste 1	Mini-teste 2	Frequência	Nota da avaliação contínua	Exame	Passou / Chumbou
2		20%	20%	60%		100%	
3	Joana Comapapa	19,2	17,4	7,1	11,58		

Function Passou_chumbou (**nota_cont**, **nota_freq**, **nota_exame**)

IF nota_cont >= 10 AND nota_freq >= 7.5 OR nota_exame >= 10 THEN

Passou_chumbou = "Passou"

ELSE

Passou_chumbou = "Chumbou"

END IF

End Sub

Exercício Extra - Do While

- Preencha as células A2:A10 com números, escrevendo um programa que lê um número e posiciona a célula activa no número lido.

```
Public Sub exercicio()
```

```
Dim num As Integer
```

```
num = InputBox("Introduza o numero: ")
```

```
Dim c As Range
```

```
Set c = Range("A2")
```

```
Do While c.Value <> num
```

```
    Set c = c.Offset(1, 0)
```

```
Loop
```

```
c.Activate
```

```
End Sub
```

Exercício Extra - Indexação

- Mostra o conteúdo das 3 células abaixo da célula activa.

```
Public Sub exemplo_indexacao()  
Dim i As Integer  
For i = 1 To 3  
    MsgBox ActiveCell.Offset(i, 0)  
Next i  
End Sub
```

Exercício Extra – Célula Activa

- Ler um número; inseri-lo na célula activa (se for maior do que o que lá está).

```
Public Sub celula_activa()  
Dim value As Integer  
Value = InputBox ("escreva um numero inteiro:")  
If value > ActiveCell.value Then  
    ActiveCell.value = value  
End If  
End Sub
```

Exercício Extra – If

- Mostrar numa janela (MsgBox) o primeiro número par maior ou igual a B1. Exemplo: Se B1 contém 12 mostrar 12; se B1 contém 13 mostrar 14, etc.

```
Public Sub mostra_par()
Dim val As Integer
val = Range("B1").value
If val Mod 2 = 0 Then
    MsgBox val
Else
    MsgBox val + 1
End If
End Sub
```

Exercício Extra – If

- Acertar no conteúdo das células K1 ou K2.

```
Public Sub acertar()
```

```
Dim S
```

```
S = InputBox("Escreva qualquer coisa:")
```

```
If S = Range("K1") Or S = Range("K2") Then
```

```
    MsgBox ("Acertou!")
```

```
Else
```

```
    MsgBox ("Errou!")
```

```
End If
```

```
End Sub
```

Exercício Extra – If

- Acertar no conteúdo das células K1 ou K2, mas modificar a subrotina: em vez de usar OR usar AND e NOT.

```
Public Sub acertar()
```

```
Dim S
```

```
S = InputBox("Escreva qualquer coisa:")
```

```
If Not (S <> Range("K1").value And S <> Range("K2").value) Then
```

```
    MsgBox ("Acertou!")
```

```
Else
```

```
    MsgBox ("Errou!")
```

```
End If
```

```
End Sub
```

Exercício Extra – If

- Faça uma subrotina que insira o “número seguinte” na célula de cima da célula activa: se a célula de cima estiver vazia insere 1; se a célula de cima contiver um número insere o número seguinte; senão, envia uma mensagem (MsgBox) a dizer que não pode fazer nada.

```
Public Sub numero_seguinte()
If ActiveCell.Offset(-1, 0).value = "" Then
    ActiveCell.value= 1
Elseif IsNumeric(ActiveCell.Offset(-1, 0)) Then
    ActiveCell.value= ActiveCell.Offset(-1, 0).value + 1
Else
    MsgBox("Não posso fazer nada!")
End If
End Sub
```