. . . . . . . . . . . . . . **PHẦN MỞ ĐẦU**

Microsoft excel là một công cụ mạnh dùng trong việc phân tích & trình bày các thông tin. Thế mạnh của bảng tính excel, ngoài các công thức sẵn có, còn là macro của nó.

Kể từ khi xuất hiện version 5.0 đến nay, ngôn ngữ dùng trong excel là Visual Basic for Applications (VBA). Đó là ngôn ngữ lập trình dùng chung cho nhiều phần mềm trong windows.

Quyễn ghi chép này sẽ nhằm giới thiệu cách thức làm quen với macro, và cách thức tuần tự chinh phục ngôn ngữ VBA trong excel.  
  
**Bài 1: Những macro đơn giản nhất**  
  
***1. Bộ thu macro***  
  
Ngôn ngữ macro là một ngôn ngữ lập trình thật sự. Nhưng mặt trái của sự mạnh mẽ này là làm cho ta không dễ đọc & hiểu ngay được.  
  
Để bước đầu làm quen với nó ta nhờ bộ thu macro, bộ thu này luôn có sẵn trong excel. Nó giúp bạn tạo ra những tác phẩm đầu tay và rất thực tế. Để rồi sau đó ta sẽ tạo ra những macro phức tạp, thực hiện những công việc theo sở thích của bạn!   
  
***2. Tạo macro đầu tay***  
  
Khi thực hành với excel, chúng ta đôi lúc cũng phải định dạng dẫy ô nào đó. Muốn dẫy biểu thị dạng tiền VNĐ không có số lẽ, ta chọn dãy ô; vô menu Format -> Cells; ->Number ta chọn Currency trong hộp thoại Category; trong hộp Decimal places ta chọn giá trị 0; trong hộp Symbol ta chọn None; cuối cùng trong hộp Negative numbers ta chọn hàng thứ ba (đang là (1.2340)) và nhấp vô nút OK để trở về trang tính   
  
Excel mặc định dạng thức tiền có hai số lẽ. Trong thực tế nhiều khi không cần nhiều số lẽ đến vậy!. Chúng ta tạo một macro để định dạng ô theo kiểu tiền không có số lẽ.  
  
a./ Chuẩn bị:  
  
Ta quét chọn các ô từ A2 đến A9; bấn chuột lên thanh công thức và nhập các hàm sau:  
  
= INT(9 \* RAND()) ^ 8 và kết thúc bằng tổ hợp hai fím CTRL+ENTER   
  
b./ Tạo macro  
  
Vô menu Tool -> Macro ->Record New Macro. . .; cửa sổ New Macro xuất hiên; Trong hộp Macro Name ta nhập FormatCurrency và nhấp nút OK.   
  
Trên màn hình sẽ hiện thêm thành phần thanh Toolbar của macro. Chúng ta bắt đầu thực hiện việc ghi vô bộ thu macro:  
  
(Với các ô A2:A9 đã chọn), vô menu Format -> Cells. . . và chọn ngăn Number; Tiếp theo chọn Currency trong hộp thoại Category; trong hộp Decimal places ta chọn giá trị 0; trong hộp Symbol ta chọn None; cuối cùng trong hộp Negative numbers ta chọn hàng thứ ba (đang là (1.2340)) và nhấp vô nút OK để trở về trang tính.   
  
Kết thúc ghi macro bằng cách nhấp vô nút close trên thanh Toolbar của macro vừa xuất hiện   
  
Chú ý: Một số thao tác có thể có nhiều cách để đạt được; thí dụ tổ hợp phím CTRL +1 sẽ nhanh chóng đưa ta đến hộp thoại Format Cells; Ta cũng có thể kết thúc ghi macro khi chọn trên menu Tool -> macro -> Stop Recording; Trong phần ghi chép này chỉ ghi một cách thức (còn các cách khác ta tự áp dụng)  
  
Tác phẩm đầu tay đã xong; Chúng ta nóng ruột muốn biết đứa con tinh thần nó hình hài ra sao & sẽ chạy nhảy như thế nào!  
  
Để xem hình hài ta vô menu Tool -> Macro; trong danh sách liệt kê nó là tên duy nhất mà ta đã ấn định: FormatCurrency; Nội dung nó như sau:

Code:

**Sub FormatCurrency()**

' FormatCurrency Macro

' Macro recorded 10/10/2006 by Quai Dang

'

Selection.NumberFormat = "#,##0\_);(#,##0)"

**End Sub**

Thực ra chỉ có ba dòng lệnh trong macro là quan trọng thôi; các dòng có dấu nháy đầu dòng là những ghi chú, nhắc chúng ta fút nó chào đời như: tên, tuổi , fím nóng để vận hành . . .  
  
c./ Chạy macro   
  
Để biết nó đi đứng ra sao ta quét chọn vài ô dữ liệu số trên trang tính; vô menu Tool -> Macro -> Macros; chọn tên macro và ấn nút Run   
  
d./ Gán tổ hợp phím tắt cho macro  
  
Vô menu Tool -> Macro -> Macros; cửa sổ Macro xuất hiện trở lại; Phía phải của cửa sô (CS), ngoài nút Run mới quen, còn có các nút khác mà ta sẽ dần làm quen. Và nút kế tiếp ta làm quen là Option; Ta nhấp vô nó để hiện lên CS Macro Option; Ta bấm chuột vô ô Shortcut key; ấn giữ phím Shift và nhấn tiếp phím 'F'. (Như vậy là ta gán cho macro FormatCurrency tổ hợp 3 fím CTRL+SHIFT+F  
  
Chú ý: Tất nhiên ta cũng có thể không giữ fím Shift; nhưng việc đó lợi bất cập hại vì dễ gây xung đột!  
  
Chú ý: Ta cũng có thể gán tổ hợp phím nóng ngay từ đầu, sau khi đặt tên: Khi đó ta cũng đã thấy CS Shortcut key trong hộp thoại Record Macro  
  
***3. Macro thay đổi nhiều thuộc tính***   
  
a./ Chuẩn bị:   
Nhập vô ô B2 chuỗi sau: 'Tổng hợp số liệu tháng 10/06'. Chọn ô B2 này;  
  
b./ Thu macro:  
  
Vô menu Tool -> Macro ->Macros . . . Trong CS Macro name nhập tên: 'DinhDangTieuDe'; Trong CS Shortcut key ta giữ phím Shift & nhấn phím 'D' và nhấn nút OK để ghi macro mới..  
  
Khi ô B2 vẫn được chọn, vô menu Format -> Cells. . .Tiếp theo vô ngăn Aligment; Trong CS Text aligment ta chọn tại cả hai CS Horizontal & Vertical đều là Center; Sau đó ấn nút OK để trở về trang tính; Ta quét chọn các ô từ B2 đến F2; Sau đó nhấp vô nút Merge and Center để tiêu đề canh giữa của các ô liệt kê. Kết thúc việc ghi macro DinhDangTieuDe bằng 1 trong các cách nêu trên.  
  
Để xem nội dung macro ta nhấn đồng thời hai phím ALT + F6 để mở CS Macro và chọn tên DinhDangTieuDe; xong rồi ấn chọn nút Edit để hiện màn hình Microsoft Visual Basic có chứa đoạn code sau:

Code:

**Sub DinhDangTieuDe()**

'

' DinhDangTieuDe Macro

' Macro recorded 10/10/2006 by Quang Duc

' Keyboard Shortcut: Ctrl+Shift+D

Range("B2").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Range("B2:F2").Select

With Selection

.HorizontalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

**End Sub**

Cái macro trước thực tế chỉ có một dòng lệnh; nhưng macro này rất nhiều dòng lệnh.  
  
c./ Bước đầu tác động lên macro:  
  
Trước tiên , ta biết các dòng có dấu nháy đơn đầu dòng lá những dòng chú thích; ta bỏ hẵn 5 dòng. (Chỉ để dòng ' Keyboard Shortcut: Ctrl+Shift+D). Nhưng cắt nó dán sau 2 dấu ngoặc đơn; của dòng Sub DinhDangTieuDe()   
  
Ta vô menu Debug và chọn thanh Compile VBAProject; Nếu chúng ta không thấy Cửa Sổ thông báo lỗi nào thì ta biết rằng VBA chấp nhận những thay đổi của chúng ta (chứng tỏ không có lỗi biên dịch.)  
  
Hơn nữa ta lại biết thêm rằng, các chú thích có thể ngay sau câu lệnh, miễn trước nó có dấu nháy đơn;  
  
Tiếp theo ta thực hiện đánh số vô các dòng lệnh và sau đó mạnh dạn bỏ các dòng lệnh từ 5 đến 11. Ta lại thực hiện kiểm lỗi biên dịch nhờ menu Debug như nêu trên; Nếu không được báo lỗi nào, ta yên tâm bấm nút lưu lại  
  
Tiếp một bước xa hơn, ta đặt dấu nháy trước các số của các dòng lệnh từ 14 đến 23. Tức là ta vô hiệu các dòng lệnh này một cách tạm thời;  
  
Thực hiện phép thử macro lần cuối, như sau:  
  
Xoá nguyên dòng 2 chứa định dạng mà macro đã thực hiện; Nhập dòng khác thay thế (VD: 'BÁO CÁO TÌNH HÌNH THÁNG TRƯỚC.' và thử chạy macro DinhDanhTieuDe; Nếu nó vẫn làm việc theo đúng như lúc chưa sửa gì là ô kê!

Code:

**Sub DinhDangTieuDe()**  ' Keyboard Shortcut: Ctrl+Shift+D

1 Range("B2").Select

2 With Selection

3 .HorizontalAlignment = xlCenter

4 .VerticalAlignment = xlCenter

12 End With

13 Range("B2:F2").Select

14 With Selection

15 .HorizontalAlignment = xlCenter

16 .WrapText = False

17 .Orientation = 0

18 .AddIndent = False

19 .IndentLevel = 0

20 .ShrinkToFit = False

21 .ReadingOrder = xlContext

22 .MergeCells = False

23 End With

24 Selection.Merge

**End Sub**

Như vậy, macro của chúng ta chỉ còn là:

Code:

**Sub DinhDangTieuDe()**  ' Keyboard Shortcut: Ctrl+Shift+D

Range("B2").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

End With

Range("B2:F2").Select

Selection.Merge

**End Sub**

d./ Đọc & diễn dịch nội dung các câu lệnh:  
  
Macro được bắt đầu bằng từ Sub (tạm hiểu là từ viết tắt của Subroutine); & dòng cuối của macro là End Sub;   
  
Macro luôn có tên, (VD ta đã có DinhDangTieuDe & FormatCurrency); Hơn nữa tên phải tuân thủ một quy luật nghiêm:  
  
Không gồm các khoảng trắng, không là các ký số ở đầu tên.. . ( việc này VBA sẽ truyền đạt cho bạn rõ, một khi bạn không tuân thủ!)  
  
Macro DinhDangTieuDe thực ra làm hai công việc:   
  
- Chọn ô B2 & định dạng canh giữa theo chiều ngang & chiều dọc;  
  
- Canh giữa đoạn văn bản trên toàn bộ các ô "B2:F2"  
  
Vì một lý do nào đó, chúng ta có thể viết hai dòng lệnh trên một hàng, nhưng phải cách nhau bằng dấu hai chấm ':'  
  
Range("B2:F2").Select: Selection.Merge   
  
Ngược lại, với câu lệnh quá dài, chúng ta có thể bố trí trên nhiều dòng; lúc này dấu ngang dưới '\_' được nhập cuối dòng trên để báo cho VBA biết rằng dòng lệnh còn tiếp theo ở dòng dưới. Ta xem xét đến ví dụ macro dùng để xếp danh sách theo tên ("C1"), như sau

Code:

**Sub SortByName()**

Columns("A:C").Select

Selection.Sort Key1:=Range("C1"), Order1:=xlAscending, Key2:=Range("A1") \_

, Order2:=xlAscending, Key3:=Range("B1"), Order3:=xlAscending, Header:= \_

xlGuess, OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, \_

DataOption1:=xlSortNormal, DataOption2:=xlSortNormal, DataOption3:= \_

xlSortNormal

**End Sub**

Như vậy, câu lệnh thứ hai trong macro trãi dài qua 5 dòng & nối kết với nhau bằng dấu gạch dưới (chú ý có khoảng trắng trước nó)  
  
Câu lệnh này được hiểu như sau:  
  
Tại ba cột A-C đã chọn (ở câu lệnh trước đó), hãy xếp theo hướng tăng dần giá trị tại cột C;  
  
Tiếp theo là xếp theo thứ tự tăng dần các giá trị tại cột A;  
  
Và cuối cùng là xếp theo cột B cũng theo chiều tăng dần ;  
  
  
Bài tập của bài 1:  
  
+ Tạo macro để xoá thanh trang bảng tính (sheet tab): Thường thanh này nằm trên thanh cuộn ngang, Ta có thể vô menu Tool-> Option -> View rồi bỏ chọn hộp Sheet Tabs  
  
+ Chuyển công thức thành giá trị: Tại trang tính để chuẩn bị ghi macro đầu tiên ta có công thức = INT(9 \* RAND()) ^ 8 tại các ô cột A; Bạn thử lập macro chuyển các ô từ A2 đến A9 này thành giá trị!  
  
+ Tại cột D, bắt đầu từ D1, D2 ta nhập tương ứng 1 & 2  
  
Tại cột E kế bên ta nhập 1 & 4; Tại cột F ta nhập F1 là 36; F2 là 100  
  
Hãy tạo macro để thực hiên các hành động sau: Chọn vùng từ D1:F2; vô menu Edit -> Copy; Sau đó chọn ô D4 và cũng vô menu Edit -> Paste Special; tại CS Paste Special ta đánh dấu hộp kiểm Transpose; sau khi dán xong ta nhấn phím ESC để bỏ vùng chọn.  
  
+ Tại trang bảng tính của VD 3; thiết lập macro ghi lại những hành động sau:  
  
Chọn ô A1, vô menu Edit -> Go to; Trong hộp thoại

Bài đọc thêm [Xem tại đây](http://www.google.com.vn/custom?hl=vi&cof=&domains=www.giaiphapexcel.com&q=ghi+macro&sitesearch=www.giaiphapexcel.com&meta=)

**Bài 2: Những macro phức tạp hơn**  
  
  
**1. Tăng đồng loạt tiền thưởng cho một danh sách**  
  
Gần cuối tháng sếp yêu cầu bạn lập danh sách tiền thưởng của toàn cơ quan, kèm theo lệnh: "Tăng 15% so với tháng trước;  
  
Nếu tháng nào sếp cũng chỉ đạo tăng hay giảm tiền thưởng so với tháng trước, thì đây là miếng đất màu mỡ để macro hoạt động;  
  
Vấn đề là bạn đã có danh sách CNV trong toàn cơ quan. Trong đó, cột F, kể từ F3 là dữ liệu tiền thưởng tháng trước. Bạn nên thu một macro để sử dụng cho các tháng sau với những hành động như sau:  
  
\* Vô menu Tool -> Macro -> Record New Macro. . . . Tại cửa sổ (CS) Record Macro ta nhập Thuong (là tên macro), xong OK  
  
\* Quét chọn từ ô G3 đến ô G99 (ô chứa người cuối cùng trong danh sách cơ quan)  
  
\* Bấm chuột lên thanh công thức và nhập dấu bằng ( '='), dùng trỏ chuột ấn vô ô F3, sau đó nhập tiếp dấu nhân '\*' và giá trị 1,15 (như ý của sếp). Kết thúc việc nhập liệu bằng hai phím CTRL+ENTER  
  
\* Vẫn những ô được chọn, ta vô menu Edit -> Copy; Sau đó thực hiện Paste Special; Trong CS Paste Special ta đánh dấu kiểm tại dòng Value  
  
\* Công đoạn cuối cùng là cắt toàn bộ dữ liệu cột G này dán lên cột F  
  
Ta thu được macro có nội dung sau (đã thu gọn các dòng lệnh ngắn):

Code:

**Sub Thuong()**

Range("G3:G95").Select

Selection.FormulaR1C1 = "=RC[-1]\*1.15"

Selection.Copy

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

Application.CutCopyMode = False: Selection.Cut Range"F").Select: ActiveSheet.Paste

**End Sub**

**2. Thực hiện việc thêm vô macro:**  
  
Macro Thuong còn những bất tiện:   
  
- Nếu cơ quan tăng người thì phải sửa lại dòng lệnh đầu;  
  
- Lượng tăng giảm tiền thưởng là theo ý chủ quan của sếp (doanh nghiệp tư nhân)  
  
Vấn đề đầu hơi dễ giải quyết, ta cứ việc ấn định thừa lên; thay vì G95 ta nhập G450, Việc này cũng chưa phải tối ưu, nhưng tại thời điểm này nên chấp nhận giải pháp. Đợi khi nào sau này trình độ ta khá lên, sẽ giải một cách căn cơ hơn!  
  
Để giải quyết vấn đề còn lại, ta thực hiện từng bước như sau:  
  
Tại dòng lệnh đầu tiên ta nhập tiếp dấu hai chấm (':'), sau đó ấn một lần phím {TAB} và thêm chuỗi: dim StrC as string; Để kết thúc, nhấn ENTER.  
  
Nếu excel sửa lại cho ta, trở thành như sau là đúng:   
  
Range("G3:G450").Select: Dim StrC As String  
  
Tại dòng trống ta vừa tạo ra, ta nhập câu lệnh:  
  
ctrc = inputbox("HAY NHAP HE SO: ")  
  
Sau khi ENTER, nhập tiếp dòng sau:  
  
strc = "=RC[-1]\*" & strc  
  
Cuối cùng ta sửa lại dòng lệnh kế tiếp trở thành:  
  
Selection.FormulaR1C1 = strc  
  
Khi đó macro có nội dung như sau:

Code:

**Sub Thuong()**

Range("G3:G15").Select: Dim StrC As String

StrC = InputBox("HAY NHAP HE SO: ")

StrC = "=RC[-1]\*" & StrC

Selection.FormulaR1C1 = StrC

Selection.Copy

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

Application.CutCopyMode = False

Selection.Cut

Range("F3").Select

ActiveSheet.Paste

**End Sub**

*Chú ý:*   
  
\* Câu lệnh đầu tiên mà ta thêm vô macro là câu lệnh khai báo với VBA rằng tôi sẽ sử dụng biến có tên là StrC, kiểu chuỗi. Câu lệnh thứ nữa sẽ là: Hãy lấy giá trị cho biến StrC là giá trị tôi nhập vô; câu tiếp dịch nôm na là: hãy nối biến chuỗi tôi vừa nhập vô sau chuỗi tôi ghi; Và cuối cùng (câu lệnh ta sửa): hãy gán giá trị biến mà tôi khai báo & thiết lập, trở thành công thức của ô hiện hành;  
  
\* Khi khai báo một tên biến, ta nên viết cả chữ hoa & chữ thường, như vậy ta tận dụng được lợi thế sẵn có của chương trình kiểm lỗi chính tả của VBA; Ở trường hợp này là StrC;  
  
Tuy là khai báo như vậy, nhưng khi nhập tên biến ta cứ nhập không theo như vậy, và nhờ VBA sẽ sửa cho ta, là một dịp để kiểm tra phát hiện lỗi chính tả do nhập sai tên biến. Tên biến cũng không được chứa các kí tự toán học & không được dài quá 64 kí tự  
  
\* Về kiểu của biến thí có rất nhiều; đến đây ta chỉ cần biết những kiểu hay thường dùng:  
  
Kiểu Byte là kiểu số nguyên, miền giá trị từ 0 đến 255  
  
Kiểu Boolean: là kiểu logic, loại này chỉ nhận một trong hai giá trị: True & False  
  
Kiểu Integer: Số nguyên, miền giá trị từ -32768 đến 32767  
  
Kiểu Long: Cũng là số nguyên có miền lớn hơn: -2.147.483.648 đến 2.147.483.647  
  
Kiểu Currency: tiền tệ;  
  
Kiểu Double: kiểu số thực, được chứa trong 8 byte ô nhớ  
  
Kiểu String: kiểu chuỗi ký tự, miền lưu giữ có thể tối đa là 65.400 ký tự  
  
\* Lệnh gán, trong ngôn ngữ VBA dùng dấu = làm lệnh gán; Ví dụ sau khi khai báo biến StrC như trên, ta có thể dùng lệnh gán: StrC = "=RC[-1]\*"; Ở đây hai dấu nháy kép chỉ để bao một chuỗi cần gán vô biến.  
  
\* Trong macro có hàm InputBox( ". . ."). Hàm này làm xuất hiện hộp thoại, nhận giá trị chuỗi mà ta nhập vô nó. Trong trường hợp này nó sẽ gán vô biến StrC giá trị mà ta nhập vô;  
  
\* Trong câu lệnh dài, chiếm 2 dòng đó, ta có thể vô hiệu hoá đoạn cuối, kể từ dấu ',' thứ hai bằng cách thêm dấu nháy trước nó; Khi đó dòng hai của câu lệnh sẽ bị tô đỏ, báo cho ta biết sự bất ổn trong nó; Sự việc là do ta đã vô hiệu hoá luôn cả dấu nối 2 dòng lệnh. Để khắc phục, ta nhập thêm dấu nháy đơn vô đầu dòng bị tô đỏ đó là được; Cuối cùng, trước khi chạy tác phẩm, ta nên vô menu Debug và chọn dòng Compile VBAProject, còn để làm chi bạn đã biết rồi.  
  
Trước khi chạy macro mới ta thực hiện việc gán phím tắt CTRL+SHIFT+T cho nó; Có thể có bạn hỏi, những điều này là bắt buộc?, không, hoàn toàn tự nguyện, nhưng nên như vậy. Cũng giống như ta không nên đi bộ bằng một chân trên vĩa hè & một chân dưới lòng đường.  
  
**3./ Tính toán các giá trị trong một cột**  
  
Ta xét trường hợp một đơn vị hành chính sự nghiệp cần lập danh sách thưởng định kỳ; Để thưởng theo hệ số đơn vị và hệ số thành tích cá nhân, người ta đã lập bảng dữ liệu gồm các trường: [Ma], [Ho], [Ten], [MDV], [XL], [TThuong] (6 cột bắt đầu từ A ).   
  
Nhiệm vụ của macro là tính số tiền thưởng của mỗi cá nhân tương ứng với hệ số cá nhân và hệ số đơn vị (tại cột 'F') với sự trợ giúp của bảng được đặt tên là HeSo. Bảng HeSo này gồm 3 cột & 5 dòng. Cột thứ hai ghi hệ số thưởng của các đơn vị; Cột thứ ba ghi số tiền thưởng cá nhân tương ứng với danh hiệu thi đua đạt được  
  
Để bắt đầu thu macro ta cũng vô menu Tool -> Macro -> Record New Macro & đặt tên macro là Tinh\_Thuong.  
  
Trên trang tính ta chọn ô F6, nơi cần xuất hiện tiền thưởng của người đầu tiên trong danh sách. Tiếp theo vô menu Insert -> Function. . .. Trong CS Insert Function vừa xuất hiện, ta nhập tên hàm là VLOOKUP và nhấn nút Go và chọn trong CS Select a function hàm VLOOKUP().  
  
Khi bảng trợ giúp hàm này xuất hiện, ta thấy dấu nhắc đang ở ô Lookup\_Value, ta nhấp chuột vô ô D6 ( chứa mã đơn vị của người đầu tiên); Trong ô Table\_array ta nhập HeSo; cuối cùng trong ô Col\_Index\_Num ta nhập số 2; Xong ta ENTER để về trang tính;  
  
Ta trỏ chuột lại lên thanh công thức; nhập dấu nhân '\*' và lại vô menu Insert lặp lại các bước trên. Duy chỉ khác là thay vì số 2 ta nhập số 3 vô ô Col\_Index\_Num;  
  
Trở về trang tính, ta chọn lại ô F6 này và dùng chức năng AutoFill để chép công thức xuống các dòng dưới; (VD: tới dòng cuối là 21)  
  
Sau đó ta chọn vùng F6:F22 và bấm lên nút AutoSum trên thanh công cụ; Excel báo cho ta là sẽ tính tổng của cột & nhập vô ô 23. Ta tán thành việc làm này và kết thúc macro như cách đã biết.  
  
Macro thu được của bạn có trùng khớp như vầy?  
  
(Thực tế ta có thể vô hiệu hoá hay bỏ dòng thứ hai đi!)

Code:

**Sub Tinh\_Thuong()**

Range("F6").Select

ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-2],HeSo,2)"

ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-2],HeSo,2)\*VLOOKUP(RC[-1],HeSo,3)"

Selection.AutoFill Destination:=Range("F6:F21"), Type:=xlFillDefault

Range("F6:F21").Select

Range("F23").Select

ActiveCell.FormulaR1C1 = "=SUM(R[-17]C:R[-1]C)"

Range("F24").Select

**End Sub**

Hãy cố đọc để hiểu nội dung của nó; Trong đó có dùng hai hàm SUM() & VLOOKUP(). Kinh nghiệm cho thấy, ta đọc từ phải sang trái thì dễ hiểu câu lệnh hơn. Vì dụ hai câu lệnh gần cuối được hiểu như sau:  
  
Chọn ô 'F23'  
  
Lấy tổng các ô từ ô thứ 17 hàng trước ô hiện hành, cho tới ô trước ô hiện hành 1 ô gán cho ô hiện hành.  
  
**4./ Chuyển động tuyệt đối & tương đối**  
  
Ta xem xét đến câu lệnh Range("F23").Select & câu lệnh Range("F24").Select  
  
Trong thực tế ta không dùng chuột ấn vô nút này, mà nhập công thức tính tổng trong ô F23 xong ta thực hiện ENTER. Kết quả của nó là đến ô F24 như máy đã ghi!  
  
Trong dòng lệnh thứ hai, hàm VLOOKUP() có đối số thứ nhất được ghi là RC[-2]; chữ C chỉ số cột, chữ R chỉ số hàng; Í nghĩa thực tế là đối số thứ nhất của hàm đang tại ô cùng hàng & phía bên trái cách 2 cột  
  
Trong câu lệnh lấy tổng, chúng ta đã nói trên; nhưng ở đây VBA lại ghi theo dạng thức tương tự như "B2:D9": góc trên trái nhất & góc dưới phải nhất của vùng chọn.  
  
Để rõ hơn ta xét đến một macro ghi lại các chuyển đông bằng bàn phím như sau:  
  
Vẫn lấy ví dụ tính tiền thưởng nêu trên, mà trong đó CSDL (cơ sở dữ liệu) gồm có 23 dòng tất cả (tại cột 'F'). Ta thu một macro (tên là DiChuyen) với những hành động như sau:  
  
Chọn ô B2; ấn đồng thời 2 phím CTRL & phím mũi tên xuống (điểm sáng sẽ tới ô B5, là ô đầu tiên trong cột có giá trị (đang chứa Ký tự 'Ho' được tô đậm)  
  
Lặp lại lần nữa, ô được kích hoạt sẽ là ô B21 (là ô dòng cuối không kề dòng chứa công thức tổng).  
  
Nếu tiếp tục ta đến được dòng cuối của trang tính!  
  
Bây chừ thì phải ngược lên thôi: ấn CTRL+ phím mũi tên lên: ô kích hoạt sẽ lại là B21;  
  
lặp lại hành động này một lần nữa sẽ là – B5  
  
Cuối cùng sẻ là ô B1 nếu ta muốn!

Code:

**Sub DiChuyen()**

Range("B2").Select

Selection.End(xlDown).Select

Selection.End(xlDown).Select

Selection.End(xlDown).Select

Selection.End(xlUp).Select

Selection.End(xlUp).Select

Selection.End(xlUp).Select

**End Sub**

Có khi macro chạy nhanh quá, không làm bạn tin!?! Muốn ghi nhận lại vị trí ô được kích hoạt khi thực hiện 1 lần ta làm theo cách sau:  
  
Đầu dòng lệnh thứ nhất ta nhập câu lệnh sau  
  
dim lJ as long: và nhấn hai lần phím {TAB} (chú í có cả dấu hai chấm ':')  
  
Để điểm chèn cuối dòng lệnh thứ hai & ấn ENTER, như vậy sẽ thêm 1 dóng trắng. ta nhập lên nó hai câu lệnh cách nhau bằng dấu hai chấm, như sau:  
  
lj = selection.row: msgbox str(Lj)  
  
(chú ý: giữa chữ selection & chữ row có dấu chấm)  
  
Ta chịu khó lặp lại chuyện này ở tất cả các dòng lệnh trước cụm từ End Sub  
  
Thực hiện biên dịch Sub như đã nêu & cho chạy lại macro  
  
Trước khi kết thúc macro sẽ đưa ra sáu hộp thoại báo cho ta biết ô kích hoạt đang là hàng thứ mấy trong trang tính.  
  
Chú ý:   
  
Ø Tại sao ta phải khai báo biến lj có kiểu là Long, mà không là Integer hay Double?  
  
Nếu khai báo chỉ là Integer thì biến không đủ năng lực để đến được dòng 65536, mà đã bị bắt lỗi!  
  
Nhưng ngược lại, nếu khai báo kiểu dữ liệu Double thì quá dư thừa & lãng fí tài nguyên. Hơn nữa không vì thế mà macro chạy nhanh hơn!  
  
Ø Hàm MsgBox() tương tự như hàm InputBox() nêu trên, chỉ có điều chiều tác động thì ngược lại, hàm sau nhận thông tin tự người dùng, còn hàm mới này thông báo cho người dùng cái gì đó mà người dùng đang thiếu thông tin!  
  
Ø Hàm Str(Num), hay hàm CStr(Num) sẽ biến đổi kiểu dữ liệu dạng số ( Byte, Integer, Long. . .) thành chuỗi kí số.   
  
Cần nhớ rằng CStr(Num) & Str(Num) có khác nhau ở nhiều trường hợp! Ví dụ   
  
Cú pháp Range( "A" & CStr(9) ).Select thì VBA hiểu;

Còn Range("A" & Str(9)).Select thì không! Tuy rằng khi biên dịch VBA không phát hiện ra lỗi này!   
  
**5./ Nối các macro**  
  
Ta có thể chép hết các dòng lệnh của macro sau cùng (chỉ trừ câu Sub DiChuyen & dòng End Sub) vô trước dòng End Sub của macro nêu đầu tiên trong bài. Điều này làm được do hai macro làm các công việc hoàn toàn khác nhau, công việc này xong không còn liên quan đến công việc sau.  
  
Một cách khác nữa là ta nhập tên macro sau vô trên dòng lệnh vừa nêu của macro trước:

Code:

**Sub Thuong()**

. . . .

. . . . .

DiChuyen

**End Sub**

Hay

Code:

**Sub Tinh\_Thuong()**

. . . .

.. . . .

**DiChuyen**

**End Sub**

tất nhiên trong mỗi trường hợp, hộp thoại ta ghi thêm sẽ đưa ra thông tin có thể khác nhau!  
  
*Bài tập của bài hai :*  
  
1./ Tạo macro xoá dòng (hay cột) chứa mẫu tin   
  
2./ Tạo macro cho ẩn hay hiện cột (hay dòng) chứa dữ liệu  
  
3./ Hãy tạo bảng tính đã có các cột [ĐG] & [SL], hãy tạo macro tính cột [TTien]  
  
4./ Hãy tạo macro xếp học lực cho HS theo điểm trung bình môn học.

**.Bài 3 Macro thực hiện các việc lặp lại**  
  
**1./ Tạo macro tô màu nền một ô**

Trước tiên ta cần chuẩn bị CSDL (cơ sở dữ liệu) như một danh sách (DS) học sinh của trường phổ thông hay một cơ quan nào đó gần ngàn người; có tối thiểu các trường dữ liệu như sau: [Ma], [Ho], [Ten], [NgaySinh], & một số trường khác nữa. . . Nhiệm vụ nêu ra là đến cuối bài học ta sẽ tô màu khác nhau cho những người trong DS có các tháng sinh khác nhau;

Bước đầu ta thực hiện việc tô màu cho 1 ô & di chuyển xuống ô kế tiếp. Ta vô menu Tool -> Macro -> Record New Macros và đặt tên macro là **ToMau**

Sau khi ấn nút OK trở về bảng tính ta chọn cột (trường) [NgaySinh] với ô trên cùng chứa dữ liệu của người đầu tiên trong DS. (Ví dụ đó là ô ‘G6’)  
Tiếp theo ta rà mũi chuột vô biểu tượng Fill Color trên thanh Toolbar, bấm hình mũi tên xuống để mở bảng màu nền.  
Ta nhấn vô ô màu trái nhất hàng dưới cùng; xong ta nhấn mũi tên xuống trên bàn phím.  
Lại rà mũi chuột lên bảng màu, ấn chọn vô màu kề nó phía phải màu vừa chọn & lại ấn phím mũi tên xuống;  
Lặp lại quá trình này cho đến hết số màu trong hàng cuối của bảng màu;  
Kết thúc thu macro, ta sẽ có nội dung như sau (tác giả đã thu gọn một số cập lệnh lên trên một dòng & bỏ bớt các câu lệnh tô màu từ ô G11 trở đi):

Code:

**Sub ToMau()**

Range("G6").Select

With Selection.Interior

.ColorIndex = 38: .Pattern = xlSolid

End With

Range("G7").Select

With Selection.Interior

.ColorIndex = 40: .Pattern = xlSolid

End With

Range("G8").Select

With Selection.Interior

.ColorIndex = 36: .Pattern = xlSolid

End With

Range("G9").Select

With Selection.Interior

.ColorIndex = 35: .Pattern = xlSolid

End With

Range("G10").Select

With Selection.Interior

.ColorIndex = 34: .Pattern = xlSolid

End With

**End Sub**

Trong macro, mỗi khi đến 1 ô mới, VBA gán 2 thuộc tính màu nền và đặt tính pattenrn solid cho ô. Macro này chỉ tiện hơn cách làm thủ công tí tẹo!

Chú ý: Hai câu lệnh gán thuộc tính cho mỗi ô có khác hơn so với macro FormatCurrency đầu tay. Ở đây tên mỗi thuộc tính đều nằm riêng biệt, mà không nằm sau từ Selection & cách 1 dấu chấm như ở Sub FormatCurrency.

Cặp câu lệnh With Selection . . . End With làm các câu lệnh nằm giữa chúng sẽ tác động lên phần chọn; coi như mỗi câu đều có từ Selection. đứng trước. Rõ ràng cấu trúc này khiến cho macro sáng sủa & dễ đọc hơn. Còn hơn thế, nó làm cho macro chạy nhanh hơn vì giảm bớt một số khâu trở lại ban đầu: Thay vì chỉ vô căn phòng 1 lần, rót nước uống, rồi châm thuốc hút, ta lại vô fòng, rót nước uống, lại ra, xong vô lại & châm thuốc hút trong fòng í!

**2./ Chỉ cho macro biết lựa chọn**  
Vấn đề nêu ra là macro phải biết tô màu nào ứng với giá trị chứa trong ô đó; Và mục tiêu của ta sẽ là tô màu khác nhau cho các tháng sinh khác nhau. Để làm việc này ta tô chọn tất cả các dòng, trừ hai dòng đầu & cuối và mạnh tay xoá cả đi (bằng phím Delete)  
Nhập các dòng lệnh dưới đây

Code:

**Sub ToMau()**

1 Dim Thang, StrC As String: Dim Ij As Integer

2 For Ij = 6 To 999

3 StrC = "G" & CStr(Ij): Range(StrC).Select

4 With Selection

5 Thang = .Value

6 If .Value = "" Then Exit For

7 Thang = Month(Thang)

8 Select Case Thang

9 Case Is < 3

10 .Interior.ColorIndex = 38

11 Case Is < 5

12 .Interior.ColorIndex = 40

13 Case Is < 7

14 .Interior.ColorIndex = 36

15 Case Is < 9

16 .Interior.ColorIndex = 35

17 Case Is < 11

18 .Interior.ColorIndex = 34

Case Else

20 .Interior.ColorIndex = 37

End Select

End With

23 Next Ij

Exit Sub: **End Sub**

Ta khai báo biến StrC để lưu giữ một chuỗi các địa chỉ thay đổi (tăng dần từ G6 trở đi)

*Chú ý:* 1. Biến Thang ở đây không được khai báo kiểu dữ liệu, lúc đó máy mặc nhiên hiểu kiểu dữ liệu là Variant. Một biến kiểu Variant có thể chứa các kiểu dữ liệu mà ta biết từ bài đầu đến nay. Tất nhiên bộ nhớ giành cho nó cũng phải nhiều hơn, & sau này nó còn chứa được nhiều thứ khác nữa.

2. Ta có thể ghi số vô đầu mỗi câu lệnh, để cho máy cũng như ta nhận biết điều này.

3./ Vòng lặp For . . .Next  
Để macro có thể thực hiện chu trình lặp lại, ta sử dụng một cấu trúc tạo vòng lặp từ câu lệnh dòng 2 đến dòng 23; Trong một lần lặp, macro sẽ làm những việc sau: Lấy giá trị trong ô để tính ra tháng sinh nhật; tô màu nền ứng với tháng tìm được.  
Trong cấu trúc này khối các câu lệnh trong vòng lặp bình thường sẽ thực hiện theo số lần đã xác định; mà cụ thể ở đây là (999 – 6) lần. Tuy nhiên tại câu lệnh số 6, người ta ấn định điều kiện nếu trong ô không chứa giá trị sẽ thoát vòng lặp. /(/hư vậy, thay vì lặp lại những tính toán vô nghĩa, máy sẽ thoát ra ngay đúng lúc. Và các bạn sẽ tiết kiệm rất nhiều tài nguyên một khi dữ liệu chúng ta đang ít.  
Trong excel chúng ta cũng đã thấy bóng dáng vòng lặp này trong hàm =FACT(Num);   
Cú pháp đầy đủ của vòng lặp **For** . . . **Next** như sau:

Code:

**For** Counter = First **To** last [**Step** step]

(statements)

[**Exit for**]

(statements)

**Next** [Counter]

Trong đó, các từ tô đậm là từ khóa của VBA, những từ khóa mà VBA giành quyền sử dụng thì ta nên tránh càng xa càng tốt, nếu không muốn bị báo lỗi xâm phạm chủ quyền.

Counter là một biến đếm, (trong VD chúng ta là ij) tùy phạm vi vòng lặp mà ta khai báo kiểu dữ liệu của biến này cho hợp lẽ.  
First là giá trị ban đầu của bộ đếm; còn last là cuối; Nhưng nên biết rằng cuối chưa chắc là lớn, vì có những người, những lúc cần đếm ngược, như

Code:

**Sub CaiSoDe()**

For iZ = 65535 to 1 Step -5

If iZ Mod 999 = 0 then Exit For

Next iz

Msgbox Str(iZ)

**End Sub**

Ở đây, lưu í trước tiên biến iZ phải được khai báo như sau: Dim iZ As Long

Còn trong ví dụ, biến iZ không được khai báo trước. Và như vậy biến sẽ được hiểu có kiểu dữ liệu Variant. (Điều không báo trước này trong nghề y khuyên là chống chỉ định!)

Đây là vòng lặp giảm dần, mỗi lần biến đếm iZ giảm 5 đơn vị cho tới 1 hay cho tới khi gặp số iZ chia hết cho 999. Mod là một toán tử, hiểu như trong excel sẽ là

=If( MOD( iZ ; 999) = 0 ; ‘Exit For’; ‘Tiếp’)

Như vậy ta cũng thấy, nếu bước không được xác định nó sẽ lấy giá trị là 1. Một chống chỉ định nữa là đừng bao giờ thay giá trị biến đếm (iZ, hay Ij . . ) khi đang còn trong vòng lặp; chuyện này nên hỏi ‘bác sỹ’ nếu chưa rành về VBA)

**4./ Cấu trúc Select. . . .End Select**

Trong excel ta đã biết cấu trúc phân nhánh bằng cách dùng hàm   
 ***=IF(logical\_test; value\_if\_true; value\_if\_false)***  
Hàm này chúng ta hiểu nôm na là trên đường lưu thông chúng ta tới ngã ba; Rẽ phải hay trái tùy thuộc vé ta cầm trên tay; Hiển nhiên, trong excel cấu trúc If được phép lồng nhau (đến 7 lần); Vậy thực ra ta hiểu: không phải trước ngã ba, mà là còn chia các ngã khác nữa.   
Lệnh Select Case sẽ được phép rẽ nhiều nhánh ngay một lúc;  
(ấu trúc cú pháp của Select Case như sau:  
**Select Case** TestExpression

Code:

[**Case** Expression (i)]

[Statements (i)]

[**Case Else**

[ElseStatements]

**End Select**

TestExpression là 1 biểu thức số hay chuỗi bất kỳ. Trong ví dụ trên biến Thang giữ vai trò này. Cần nói thêm rằng lúc đầu Thang chứa giá trị dữ liệu của ô hiện hành (dòng 5); sau đó Thang chứa kết quả =MONTH(Thang) – Được gán giá trị tháng của dữ liệu (dòng 7) & là TestExpression của cấu trúc Select Case. Trong macro ToMau nêu trên, chúng ta chỉ tô 6 màu khác nhau cho từng cặp tháng; tuy nhiên chúng ta cũng có thể viết

Code:

Select Case Thang

Case 1, 7

.Interior.ColorIndex = 38

Case 2, 8

.Interior.ColorIndex = 40

. . . .

Case Else

.Interior.ColorIndex = 37

End Select

Những người thành thục VBA khuyên ta nên luôn nhập câu lệnh Case Else cho mọi trường hợp, có khi sẽ chộp được những giá trị TestExpression không mong muốn xuất hiện.  
**Bài tập thực hành:**

1./ Hãy tạo macro tô màu cho Font chữ trong các ô tương ứng là Đỏ, Vàng, lục, lam, tím. . .; xem & đọc nội dung macro để thực hiện các bài tập dưới;

2./ Hãy đọc để biết macro sau cho kết quả bao nhiêu, khi ta nhập 9 vô hộp thoại xuất hiện:

Code:

**Sub TongBinhPhuong()**

Dim jZ as integer, wZ as Integer, TongBF As Long

wZ =InputBox(“HAY NHAP SO CUOI”)

TongBF = 0

For jZ = 1 To wZ

TongBF = TongBF + jZ \* jZ

Next jZ

Msgbox Str(TongBF)

**End Sub**

3./ Giả sử ta có cột “D” là điểm trung bình của học viên trong lớp; ta tạo macro để tô màu theo xếp loại học sinh; (VD: >=9: màu đỏ; >=7,5: Màu vàng . . .)

Bài đọc thêm: [Colors](http://www.google.com.vn/custom?domains=www.giaiphapexcel.com&q=colors&sitesearch=www.giaiphapexcel.com)

Option Explicit

**VI. Tạo bảng màu, tên màu & chỉ só của 56 màu**

**Sub colors56()** '57 colors, 0 to 56

Const Cot = 5: Const Hang = 1

Application.ScreenUpdating = False

Application.Calculation = xlCalculationManual 'pre XL97 xlManual

Dim iZ As Long

Dim str0 As String, str As String

For iZ = 0 To 56

Cells(iZ + Hang, 1 + Cot).Interior.ColorIndex = iZ

Cells(iZ + Hang, 1 + Cot).Value = "[Color " & iZ & "]"

Cells(iZ + Hang, 2 + Cot).Font.ColorIndex = iZ

Cells(iZ + Hang, 2 + Cot).Value = "[Color " & iZ & "]"

str0 = Right("000000" & Hex(Cells(iZ + 1, 1 + Cot).Interior.CoLor), 6)

'Excel shows nibbles in reverse order so make it as RGB

str = Right(str0, 2) & Mid(str0, 3, 2) & Left(str0, 2)

'generating 2 columns in the HTML table

Cells(iZ + Hang, 3 + Cot) = "#" & str & "#" & str & ""

Cells(iZ + Hang, 4 + Cot).Formula = "=Hex2dec(""" & Right(str0, 2) & """)"

Cells(iZ + Hang, 5 + Cot).Formula = "=Hex2dec(""" & Mid(str0, 3, 2) & """)"

Cells(iZ + Hang, 6 + Cot).Formula = "=Hex2dec(""" & Left(str0, 2) & """)"

Cells(iZ + Hang, 7 + Cot) = "[Color " & iZ & "]"

Next iZ

done:

Application.Calculation = xlCalculationAutomatic 'pre XL97 xlAutomatic

Application.ScreenUpdating = True

**End Sub**

**VII. Hàm trả về các dạng biểu thị chỉ số màu nền của ô được chỉ định**

**Function ShowColor(rRange As Range, Loai As String)**

Dim sColor As String

sColor = Right("000000" & Hex(rRange.Interior.CoLor), 6)

sColor = Right(sColor, 2) & Mid(sColor, 3, 2) & Left(sColor, 2)

Select Case UCase$(Loai)

Case "H"

ShowColor = sColor

Case "I"

ShowColor = rRange.Interior.ColorIndex

Case "F"

ShowColor = rRange.Font.ColorIndex

Case "T"

ShowColor = "#" & sColor

Case Else

End Select

**End Function**

**VIII. Các hàm tính toán trên cơ sỏ màu nền của các ô**

**Function ColorFunction(ColorCell As Range, rRange As Range, Optional TuyBien As** **String)**

Dim vResult, iCell As Range: Dim iIndex As Long, Dem As Long

'Written by Ozgrid Business Applications

'Sums or counts cells based on a specified fill color.

If TuyBien = "" Then TuyBien = "T"

iIndex = ColorCell.Interior.ColorIndex

For Each iCell In rRange

If iCell.Interior.ColorIndex = iIndex Then

Dem = 1 + Dem

vResult = WorksheetFunction.SUM(iCell, vResult)

End If

Next iCell

Select Case UCase$(TuyBien)

Case "D"

vResult = Dem

Case "V"

vResult = vResult / Dem

Case Else

End Select

ColorFunction = vResult

**End Functd9i5I**

**Sub DoiMau()**

Color\_Change Selection

**End Sub**

**IX. Tô màu tương ứng cho các ô theo giá trị của ô:**

**Private Sub Color\_Change(ByVal Target As Range)**

Dim rgArea As Range, rgCell As Range

Dim iColor As Integer

' Get the intersect of the target & the proper range

Set Target = Intersect(Target, Range("A11:D28"))

If (Not Target Is Nothing) Then ' If this intersection exists

For Each rgArea In Target.Areas ' For each subsection of the selection

For Each rgCell In rgArea.Cells ' For each cell of the subsection

If rgCell.Value < 56 And rgCell.Value > 0 Then

rgCell.Interior.ColorIndex = Int(rgCell.Value)

Else

rgCell.Interior.ColorIndex = xlNone

End If

Next rgCell, rgArea

End If

**End Sub**

**X. Tìm màu nền tương ứng với màu Font**

**Sub RealInvertColors()**

Dim Rng As Range

Dim reD As Double, bLue As Double, gReen As Double, CoLor As Double

Sheets("S2").Range("A20").Select

Set Rng = Selection

CoLor = Rng.Font.CoLor: MsgBox str(CoLor), , "Font Color:"

reD = CoLor Mod 256: MsgBox str(reD), , "RED Color:"

CoLor = (CoLor - reD) / 256: MsgBox str(CoLor), , "(Color - RED)/256:"

gReen = CoLor Mod 256: MsgBox str(gReen), , "Green Color:"

bLue = (CoLor - gReen) / 256: MsgBox str(bLue), , "Blue Color:"

reD = 255 - reD

gReen = 255 - gReen

bLue = 255 - bLue

' CoLor = 255 \* 255 \* blue + 255 \* green + red

' MsgBox str(CoLor)

Selection.Interior.CoLor = RGB(reD, gReen, bLue)

**End Sub**

**XI. Tìm các ô chứa giá trị chuỗi "JjWwZz"**

**Sub SelectJjWwZz()**

Dim RgJjWwZz As Range, RgNext As Range, FirstAddress As Range

With ActiveSheet.Cells

Set RgNext = .Find(What:="JjWwZz", After:=Range("A1"), LookIn:=xlValues)

If Not RgNext Is Nothing Then 'Neu Tim Thay

Set FirstAddress = RgNext

Set RgJjWwZz = RgNext

Do

Set RgNext = .FindNext(RgNext)

Set RgJjWwZz = Union(RgJjWwZz, RgNext)

Loop While RgNext Is Nothing Or RgNext.Address <> FirstAddress.Address

End If

End With

RgJjWwZz.Select

**End Sub**

**XII.Tim "Jn" trong các tên cuả WorkBook , màu đỏ thì đổi thành trắng**

**Sub HighLightNames()**

Dim Jn As Name

On Error Resume Next

For Each Jn In ThisWorkbook.Names

If Not Range(Jn).Interior.ColorIndex = 3 Then

Range(Jn).Interior.ColorIndex = 3

Else: Range(Jn).Interior.ColorIndex = 0

End If

Next Jn

On Error GoTo 0

**End Sub**

**XIII. Các bạn tự tìm hiểu :**

**Sub PhAn()**

Dim StrC As String, FirstAddress As String

Dim uRange, Jz As Integer

StrC = InputBox("HAY CHON FUONG AN:")

With Worksheets("S2").Range("A2:C25")

Select Case UCase$(StrC)

Case "B" 'Blanks: Count

Set uRange = Cells.SpecialCells(xlCellTypeBlanks)

If Not uRange Is Nothing Then

FirstAddress = uRange.Address

Do

Jz = Jz + 1

Loop While Not uRange Is Nothing And uRange.Address <> FirstAddress

End If

Case "C" 'Consts: Count

Set uRange = Cells.SpecialCells(xlCellTypeConstants, 23)

If Not uRange Is Nothing Then

FirstAddress = uRange.Address

Do

Jz = Jz + 1

Loop While Not uRange Is Nothing And uRange.Address <> FirstAddress

End If

Case "F" 'Formulas => Value 5

Set uRange = Cells.SpecialCells(xlCellTypeFormulas, 23)

If Not uRange Is Nothing Then

FirstAddress = uRange.Address

Do

uRange.Value = 5

Set uRange = .FindNext(uRange)

Jz = Jz + 1

Loop While Not uRange Is Nothing And uRange.Address <> FirstAddress

End If

Case "T" 'Find Value= 5 => '=A20'

Set uRange = .Find("5", LookIn:=xlValues)

If Not uRange Is Nothing Then

FirstAddress = uRange.Address

Do

uRange.Value = "=$A$20"

Set uRange = .FindNext(uRange)

Jz = Jz + 1

Loop While Not uRange Is Nothing And uRange.Address <> FirstAddress

End If

End Select

MsgBox FirstAddress, , str(Jz)

End With

**End Sub**

**Bài 4 Macro nhập liệu vô CSDL (cơ sở dữ liệu).**

**1./ Tạm hiểu vể CSDL & cách tìm đến dòng cuối CSDL**

Trong thực tế ít nhiều chúng ta đều tiếp xúc với CSDL. Trong excel ta có một số hàm CSDL, ví dụ =DSUM(database,field,criteria) ; database phải thỏa mãn một số yêu cầu:

Hàng đầu tiên là các tên trường, Các tên trường, cũng như tên biến: không có chứa khoảng trắng hay các kí tự đặc biệt, Các hàng trong CSDL (gọi là các record) đều phải có số liệu. Không được dùng ký tự ntr, hay ‘–‘ để thay cho các trị trùng với giá trị trường đó của record trên nó.

Ta có thể thấy CSDL học sinh các lớp trong một trường học, DS (danh sách) công nhân viên chức trong đơn vị, bảng kê hóa đơn mua bán hàng hóa, bảng kê sản lượng của bộ phận sản xuất trong đơn vị & còn nhiều những DS khác nữa . . . .

Giả sử ta có DS các chuyến hàng nhập vô nhà máy, có các trường như sau: [STt], [Ngay], [MKH], [SoXe], [TgLuong], [Bi], [Sluong]. Hàng ngày cơ sở sản xuất ấy nhận vô khoảng vài chục xe hàng để làm nguyên liệu. Như vậy sổ ghi chép đầy đủ số liệu nhập loại nguyên liệu này trong thời kỳ nào đó sẽ là một CSDL tốt để chúng ta dùng macro tác động đến tất cả các khâu, từ khâu nhập liệu, thống kê, xử lý số liệu trong báo cáo, quản lý & điều chỉnh quá trình. . .

Nếu không có công cụ macro, chúng ta phải nhập trực tiếp các số liệu của xe hàng vô dòng cuối của CSDL. Làm như vậy rất dễ nhầm lẫn, dẫn đến những sai sót khôn lường. Cách tốt nhất đến thời điểm này (do ta chưa biết gì về Form) là ta nhập số liệu về xe hàng mới vô lên 1 sheet (có tên là Nhap) theo cột từ trên xuống. (Bố trí theo cột vì thường nhập xong số liệu một ô, ấn ENTER thì con trỏ xuống ngay ô dòng dưới!). Sau khi kiểm tra xong, ta ấn nút để macro thay ta chép các số liệu này đến Sheet chứa CSDL (có tên là CSDL), vào đúng nơi yêu cầu: dòng cuối của CSDL. Lúc đó ta tận dụng được cách mà excel copy chuyển cột thành hàng;

Giải bài toán nêu trên bằng cách: Tại Nhap ta thiết kế như sau: Trộn 2 ô A1 & B1 để nó chứa chuỗi: ‘Nhập mới’; Các ô trong cột A kể từ A2 trở xuống chứa lần lượt các chuỗi: ‘Số TT’, ‘Ngày nhập’, ‘Mã chủ hàng’, ‘Biển số’, ‘Trọng lượng cả bì’, ‘Trừ bì’, ‘Số thực nhập’

Ta nhập số liệu một xe hàng vô cột b từ B2 đến B7; Còn giá trị tại B8 ta cài công thức = (B6 – B7). Đến lượt các bạn hãy tự tạo cho mình một macro chép số liệu tại cột B từ B2 cho đến B8 theo các bước sau:

v Đặt tên macro sắp thu. . .

v Dùng chuột tô chọn các ô từ B2 trở xuống B8 của sheet ‘Nhap’, xong vô menu Edit -> Copy;

v Ta chọn tiếp sheets ‘CSDL’; ấn chọn ô A2, & lại vô menu Edit -> Paste Special; Trong ngăn Paste của hộp thoại Paste Special ta chọn Values & ấn vô nút chọn Transpose

v Kết thúc thu macro, & macro này bạn đặt tên Nhap (hay NhapLieu) hay tên nào khác gợi nhớ mà bạn muốn. (Tác giả quyễn ghi chép này muốn các bạn theo dõi bài cần thực hiện macro này tối thiểu 2 lần, nên mới gợi tên tại đây; dù sao cũng là một ý đồ!) Sau mỗi lần thử ta có thể xóa macro đi bằng cách: Tại CS (cửa sổ) excel ta nhấn cùng lúc ALT+F8, CS Macro mở ra, ta chọn tên macro trong ngăn Macro Name mà ta muốn xóa, xong chọn nút lệnh Delete để thực hiện việc này. Mình khuyên các bạn nên xóa tác phẩm macro của mình nhiều lần, sau mỗi lần xóa macro, kiến thức VBA của ta sẽ lớn lên một ít!.

(/iệc này không phải là quên, mà là có chủ tâm: sau bước hai nêu trên, bạn còn một công đoạn nữa là ấn nút ESC trên bàn phím! Việc này ta có thể quên trong khi thao tác trên bảng tính excel (để bỏ chọn dãy ô định Copy). Nhưng ở đây không nên quên tẹo nào, vì dễ bị phiền phức & kiện cáo về sau, hoạc giả chúng ta không biết macro dẫn ta đi đến phương trời nào nữa?!

Chú ý:

\* Trong hộp thoại Paste Special bạn thử không chọn nút ấn Value 1 lần và xem macro cho kết quả ra sao?

\* Bạn thử một lần đặt tên macro là Nhap\_ sẽ bị VBA phản đối hay không ?!

\* Bạn nghĩ xem còn có thể làm gì để giao lưu giữa bạn & VBA thông qua macro này không? VD như thử đặt tên là \_Nhap xem bị phản đối không?! . . .

Đến đây là xong một công đoạn. Phần thứ đến là xác định dòng cuối của CSDL; Để thực hiện việc này bạn cần tự tạo cho mình một CSDL khoảng vài chục records, và na ná như mình đề ra để dễ theo dõi tiếp. Macro trên bao giờ ta cũng chép vô range A2; Nhưng thực tiển CSDL tăng (giảm) liên tục, nên ta phải làm sao để macro thông minh, hiểu ý ta mà tìm đến dòng cuối để dán dữ liệu vô!

Ta lại thực hiện ghi macro Sub DongCuoi() bằng cách thu macro qua các bước sau:

\* Chọn ô A1 của CSDL, giữ phím CTRL & ấn phím mũi tên xuống (Dòng cuối của CSDL được chọn)

\* Tiếp tục thực hiện hành động như vậy đễ đến được dòng cuối của trang tính

\* (Cuối rồi nên quay lại) Ấn tổ hợp CTRL+ phím mũi tên lên để trở lại dòng cuối của CSDL.

Sau khi ngưng thu macro ta sẽ có những dòng tương tự vày trong CS chứa các macro:

Code:

**Sub DongCuoi()**

Rang(“A1”).Select: Selection.End(xlDown).Select

Selection.End(xlDown).Select: Selection.End(xlUp).Select

**End Sub**

Điều chúng ta cần là máy cho ta biết dòng trống kế tiếp để chép; điều này thực hiện được bằng cách sửa lại macro trên để có nội dung sau:

**Sub DongCuoi()**

Dim iRow As Long

Range("A65535").Select

Selection.End(xlUp).Select

iRow = 1 + Selection.Row

MsgBox Str(iRow)

**End Sub**

Trong macro này có 2 dòng lệnh đáng kể đến. Đó là dòng lệnh thực hiện thao tác CTRL+ mũi tên lên. Dòng lệnh này làm ô chứa dữ liệu cuối của CSDL trên cột ‘A’ được kích hoạt.

Dòng thứ đến là phương thức gán cho biến iRow đã khai báo giá trị, bằng với gía trị hàng hiện hành cộng với 1 (Nên đọc dòng lệnh từ phải qua trái, là: Dòng hiện hành cộng 1, gán vô biến iRow).

**2./ Macro nhập liệu**

Đã đến giai đoạn lập macro nhập liệu, kết hợp từ hai macro nêu trên & thêm gia vị vô cho món macro thêm hấp dẫn. Nhưng giờ chúng ta dịch ngược từ ngôn ngữ macro sang ngôn từ diễn tả các bước tiến hành của ta, như sau:

Code:

**Sub Nhap()**

401 Sheets("Nhap").Select: Range("B2:B8").Select

402 Selection.Copy

403 Sheets("CSDL").Select

Range("A65535").Select

404 Selection.End(xlUp).Select

405 Selection.Offset(1, 0).Select

406 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, \_

SkipBlanks:=False, Transpose:=True

407 Application.CutCopyMode = False: Sheets("Nhap").Select

**End Sub**

Phần dịch câu lệnh:  
401: Chọn vùng cột B tại sheet (“Nhap”);  
402: Copy (vùng chọn);  
403: Tương tự 401;   
404: Đã giải thích trên;  
405: Hãy chọn ô có số hàng tăng 1 & số cột vẫn như cũ so với ô kích hoạt;  
406 & 407: ‘Quá quen!’

(Vì là ngôn ngữ dịch nên mỗi người có cách dịch khác nhau, có lẽ mình chưa thể dịch tiếng nào ra tiếng ý, mong được cảm thông!). Và các bạn thấy đấy, dịch từ ngôn ngữ VBA sang ngôn ngữ đời thường cũng quan trọng không kém, phải không?!

Nhân dịp rãnh rỗi các bạn thử bỏ vài vế sau của mệnh đề 406 xem ra răng?!

**3./ Tạo nút lệnh cho một macro**

Các bài đầu ta đã biết gán tổ hợp phím tắt cho một macro. Nhưng tiện thì có tiện với ta thôi. Còn khi ta vắng, người khác thay ta nhập liệu thì không biết các phím này!. Để tiện cho việc sử dụng ta thiết lập một nút lệnh trên trang tính, & như vậy bất kỳ ai cũng có thể xài được khâu nhập liệu.

Để làm được điều này, tại trang bảng tính ‘Nhap’ hiện hành, ta vô menu View ->Toolbars -> Forms để hiện thanh công cụ Forms. Rà mũi chuột lần lượt lên các nút để tìm nút có chữ ‘Button’. Ấn chọn nó, đưa mũi chuột đang có dạng chữ thập mảnh mai đến ô trên vùng C2:D8 (VD tôi chọn C7). Ấn giữ trái chuột & vẽ hình chữ nhật lên ô chọn. Nếu xuất hiện CS Assign Macro thì ta chọn tên macro Nhap tại Macro Name & nhấn OK.

Với Button đang được kích hoạt, ta phải chuột vô nút lệnh để hiện menu tắt của nó. Ta chọn dòng cuối là Format Control. CS Format Control xuất hiện, nhưng ta bấm Cancel cho ẩn đi. Con trỏ chuột trên dòng ‘Button n’. Ta tô toàn bộ chữ này & nhập vô thay bằng chữ việt: ‘Nhập’; Lại tô sáng chữ này & nhấp vô nút có chữ B trên Toolbar, làm đậm chữ. Sau đó chọn màu Font cho nó theo ý bạn. Nếu bạn đã, hay sẽ định gán tổ hợp các phím CTRL+SHIFT+N cho macro Nhap thì lúc này là tốt nhất để bạn tô chữ cái ‘N’ & nhấn vô nút Underline. Động tác này gây ấn tượng giữa hai thực thể ta với máy!

Sau khi tạo ra tác phẩm đầu tay này, nhất thiết việc đến tiếp là thử. Bạn có thể nhấn vô nút này nhiều lần để xem ‘Hắn’ chép những gì có trên cột ‘B’ qua CSDL; Cứ bị nhấn là chép, bất kể miễn còn được cung cấp nguồn điện!

**4./ Tạo mộtCombo Box để nhập mã khách hàng**

/(/hư trên ta thấy dòng thứ ba trong sheets ‘Nhap’ chứa mã #h (khách) hàng. Bạn có thắt mắc không? Ta không nhập toàn tên #h hàng mà chỉ là mã vì chúng ta ‘lười biếng’?

/(hông đâu các bạn! Bạn thử hình dung có nhiều #h hàng thân thiết ngày nào cũng cung cấp hai đến bốn chuyến thì người nhập cũng mệt & máy cũng mệt. Người mệt thì dễ hiểu rồi! Máy cũng mệt vì CSDL ta phình nhanh quá thay vì trường [MKH] chỉ gồm tối đa 3 kí tự hay ký số ta lại nhập nguyên Công Tằng Tôn Nữ Nguyệt Nga là không cần thiết.   
Bạn thử hình dung như thế này trong file nhân sự của một cơ quan 750 người gồm 15 bộ phận

Nếu trường [BoFan] ta nhập nguyên tên đơn vị như: Tổ chức, Hành chánh, Tài chính, PX Nguyên liệu, PX Cơ điện. . . thay cho việv nhập A, C, D, E, F . . . thì lãng phí nguồn nhân lực là đáng kể.

Tất nhiên ở đây, cũng như ở trên ta phải có bảng đối chiếu như trong hàm VLOOKUP() hay HLOOKUP() trong excel đã gặp

Trở lại với phần nhập mã #h hàng. Như vậy người dùng sẽ đưa câu chất vấn: Làm sao tôi nhớ mã của trên đôi chục khách đây?! Đúng là một yêu cầu chính đáng mà người biết macro phải đáp ứng. Và cứu tinh đó chính là Combo Box (sẽ viết tắt là Combo)

/(/ó ở đâu?, vẫn trong toolbar Form ý thôi! Bạn rà mũi chuột như ban nãy, nhưng thay vì tìm chữ ‘Button’ ta tìm chữ ‘Combo’ trong khi Tool tips hiện chữ, hay thấy cái nút nào có biểu tượng giông giống cái hộp diêm thì nhào vô kiếm.

/(/hưng kiếm vậy thôi, ta chưa chuẩn bị gì nguồn lực cho nó thì nó cũng vô dụng mà thôi; Giống như ta phải có macro Nhap rồi mới làm nút lệnh cho nó trên trang tính ấy mà!

Để chuần bị nguồn cho Combo, cũng như CSDL quản lý các thượng đế ta cũng sẽ tạo các trường như STT, Ma, HoTen, NgSinh, Dthoai, DiaChi, Fax, NguoiGD, GhiChu, . . .

Các trường khác ta bổ sung sau, nhưng hai trường phải có trước là Ma & HoTen; Giả dụ ta đã có DS này gồm 20 vị, tùy thuộc vô số lượng mà mã nên 2, 3 ký tự (Nếu dưới 1.000 thượng đế ta chỉ dùng 2 ký tự là đủ,)

Giống như tạo nút lệnh cho macro, ta cũng ấn vô biểu tượng Combo trên Toolbar Forms. Sau đó trên trang tính ta vẽ hình thanh dài đủ thấy tên của thượng đế trong đó (khoãng chừng 8 Cm). Ta cũng phải chuột vô hình chữ nhật này & chọn Format Control. Trong CS Format Control ta ấn chuột vô hộp Input range. Dùng chuột quét từ đầu đến cuối DS #h hàng. Tiếp theo, trong hộp Cell Link được ấn chuột, trên trang tính ‘Nhap’ ta chọn ô C4 (kề với ô cần nhập mã #h hàng). Tiện tay ta đánh dấu kiểm vô 3-d Shading;

Ta nhấp vô ô trống bất kỳ để kết thúc sơ lược phần format Control. Nếu giờ ta nhấn vô mũi tên xuống của Combo ta chọn tên 1 người thì tại ô C4 xuất hiện số, nói lên thứ thự của thượng đế đó trong DS. Vậy chỉ còn bước cuối cùng là ta phải liên kết giữa số trên C4 với mã #h hàng tương ứng; Điều này các bạn có thể phải tự làm lấy, mình chỉ gợi ý hai cách sau:

Tại ô C3 dùng =VLOOKUP() hay dùng hàm =CHOOSE(). Hàm trước cho DS nhiều & sau cho không tới chục #h hàng!  
Đến đây ta có thể xoa tay & cười tươi với thành quả của mình. Để khuyếch trương chiến tích, ta làm đẹp Combo bằng cách phải chuột vô Combo & thực hiện chỉnh sửa kích cỡ cho vừa ý (đối với nút lệnh cũng vậy); & tự cho phép mình tìm hiểu các CS có trong Format Control.

**Bài tập của bài 4:**

*1./ Bạn đã gặp macro này ở đâu?*

Code:

Sub Nhap\_()

Sheets("Nhap").Select: Range("B2:B8").Select

Selection.Copy

Sheets("CSDL").Select: Range("A2").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=True

Application.CutCopyMode = False

End Sub

*2./* Tại ô nhập ngày tháng (B3) ta có thể cài mặc định ngày trước ngày hiện hành 1 ngày (nếu là các thứ #) & trước 2 ngày nếu là thứ hai được không?!

*3./* Như câu hai nếu là bình thường, nhưng nếu người nhập liệu muốn lùi 1 vài ngày có được không?! (để thích ứng với vài 3 ngày nghỉ lễ trở lên í!). Gợi ý: Số ngày lùi được nhập vô C3

**File Bai4\_5.XLS là của bài 5; Nhưng các bạn có thể xem phần nhập liệu. Tham khảo thôi chứ khác nhiều so với bài 4 (để trách thắc mắc không cần thiết !)**

. . . . . . . . . . . .

**Bài 5 Macro lập báo cáo hàng ngày.**

**1./ Khảo sát các yêu cầu:**

Bài toán nêu ra là việc quản lý chu trình sản xuất tại một CS (cơ sở). Mà cụ thể là Giám đốc CS yêu cầu phân xưởng II gởi BC (báo cáo) số liệu hàng ngày trước 8h30 ngày hôm sau. Trong BC phải đủ số liệu về 4 vấn đề: công nghệ, sản lượng, chất lượng & tiêu hao;

Trong mỗi vấn đề có những yêu cầu chi tiết các hạng mục như sau:  
Cônghệ: số hư hỏng ở bốn khâu trong phân xưởng (K1, K2, K3, K4)  
Sản lượng: Sản lượng sản xuất 9 mặt hàng của p/x (phân xưởng)  
Chất lượng sản phẩm: Số loại I & & loại 2   
Tiêu hao : Điện, nguyên liệu & nhiên liệu. . . . . .(xem trong Sheet(“BCao”) **File Bai4\_5.XLS)**

Ta đã biết rằng macro nói riêng (hay VBA nói chung) rất ưa các công việc lặp lại thường xuyên này. Vấn đề là làm như thế nào để lập CSDL (cơ sở dữ liệu) xử lý một cách trơn tru & còn có hướng cho mai sau phát triển!  
Trong Sheet(“CSDL”) của file đính kèm có các trường sau: Ngay, Ca, Kip, ThSo, ThNm, SoLieu  
Ngay Ca Kíp ThSo ThNm SoLieu   
23/09/2006 A 1 B6A 28 1,230   
1/10/2006 A 1 A1A 29 9   
  
Trong đó có hai trường cần nói tới (còn các trường khác chắc ai cũng ngẫm ra). Đó là [ThNm] & [ThSo]  
[ThNm] là trường biểu trưng cho năm & tháng của [Ngay]. Sau này các bạn sẽ biết nó dùng để lọc dữ liệu theo tháng nào đó trong năm, phục vụ trong chuyện tính lũy kế sản xuất.

[ThSo] gồm 3 kí tự hay kí số. Kí tự đầu biểu trưng cho các vấn đề cần quản lý: Cônghệ, sản lượng, chất lượng & tiêu hao;

Ký số thứ đến biểu trưng cho thứ tự yêu cầu chi tiết; Ví dụ: B6A thì ta hiểu là trường SoLieu ghi sản lượng mặt hàng mác 6 (xem trong Sheets(“Nhap”))  
A1?: Ghi số hư hỏng tại công đoạn K1 của phân xưởng.  
Ký tự cuối cùng chỉ ra ca sản xuất.   
Có thể bạn sẽ hõi, vậy lưu kíp thợ làm gì? Xin thưa: Ca đêm được thêm lương & bồi dưỡng giữa ca! Việc này cần thiết để sau này lỡ Sếp trã lương theo kết quả sản phẩm cuối cùng . . .  
Cũng đúng nếu ta nói [Ca] là dư. Nhưng ta mới chập chững vô VBA nên phần sử dụng tài nguyên còn chừng mực thôi, chưa phải là tối ưu, mong nhiều bạn thông cảm!

**2./ Trích lọc số liệu ngày cần thiết**

Các bạn hãy mở trang tính ‘BCao”, trong í đã cài sẵn công thức hết rồi. Công việc của chúng ta là viết macro gán vô nút lệnh ‘Báo cáo” trong sheets ‘Nhap’. Để nó làm tuần tự các nhiệm vụ:

a.) Xếp CSDL theo [Ngay]; lí do: đề phòng nhân viên nhập liệu không theo ngày tháng như bình thường. Việc này các bạn thử ghi lại bằng một macro riêng, xem như một bài tập bổ sung!

b.) Trích lọc toàn bộ dữ liệu trong ngày thể hiện lên hai cột ‘L’ & ‘M’ trong sheet ‘CSDL’. Ngày chỉ định được ghi tại ô B2 của Sheet ‘Nhap’ hay ô H2 thuộc vùng Criteria khi ta sử dụng phương thức Advanced Filter. . trên menu Data. Các bạn cũng nên ghi 1 macro này nữa để hiểu thêm về Advanced Filter để có trong tay những gì ngôn ngữ VBA nói đến chuyện Advanced Filter. (Chuyện này không ai cấm cản bạn thực hiện nhiều lần cả, chỉ có điều bạn có muốn hay không mà thôi! Càng làm nhiều máy nó càng cảm ơn đó; Cụ thể nha: Trong menu Insert ->Name->Define của excel có ghi nhận công lao của bạn đó!) Về phần mình đáng ra tôi nên khóa các macro sẵn có trong file đính kèm, nhưng không vì thế mà các bạn không tự mình viết nhiều lần hai macro đã nêu trên (về xếp trật tự CSDL & Advanced Filter. . .) . Ý tôi muốn là các bạn tự nhận thức là chính & cơ bản nhất. Và đừng tự ái khi máy chỉ cho ta thứ này thứ khác qua ngôn ngữ VBA. Càng không nên nhào ngay vô cái sẵn có của người khác mà mình chưa có đồng vốn nào lận lưng. Và ý cuối cùng: máy tính lúc nào cũng sẵn lòng giúp ta, không hề vụ lợi!  
Nếu chúng ta thực hiện xong macro thứ hai này thì trang BC chỉ còn chưa cập nhật số liệu lũy kế tháng (nêu tại cột ‘P’ trong Sheet ‘BCao’). Còn các cột trước nó đã ngon lành.  
À còn điều này nói trước: Những bài ghi chép này là nói về VBA, nếu có ai thắc mắc về công thức nào đó của excel (mà cụ thể là trong trang ‘BCao’) thì bản thân xét thấy không có đủ khả năng giải đáp, thông cảm nha!

c.) Tạo số liệu sản lượng lũy kế tháng: Đây là công việc nhọc nhằng, vì chỉ một chục số liệu nhưng lại phải đổ nhiều công sức. Vây nên chuyển nó thành đề mục riêng luôn dưới đây:

**3./ Thu thập số liệu các ngày trước đó của tháng**

Để lấy dữ liệu từ đầu tháng từ kho lưu trữ ta thực hiện lọc hết dữ liệu từ ngày 1 của tháng đó cho đến ngày sếp yêu cầu. Nên nhớ rằng có lúc cuối tháng, nhưng sếp muốn xem kết quả sản xuất của những ngày trong tuần đầu tháng thì sao?!   
Vì chúng ta mới quen với VBA, nên chúng ta nên xài cột phụ, đó là trường [ThNm]; Nôm na là trường chứa 2 ký tự, ký tự đầu biểu trưng cho năm, ký tự sau biểu trưng cho tháng của năm ấy! Ở đây người ta làm vậy để chúng ta dễ xài mà thôi. Này nha: nếu chúng ta cung cấp cho macro một biến chuỗi kiểu ‘10/06’, ‘9/06’. . . cho máy cũng sẽ sẫy ra trường hợp bất đồng ngôn ngữ không ít thì nhiều. Vì sẽ có người cho là heo lớn hơn lợn, cũng sẽ có người cho lợn nhớn hơn heo!  
Trường này do hàm tự tạo đem lại; Để có khái niệm về hàm tự tạo, ta bắt đầu tư hàm của excel. Hàm excel luôn trả về cho ta một thứ gì đó; Có một số hàm đòi chúng ta cung cấp các đối số, nhưng cũng có những hàm ta ới là nó cho ngay mà chẵng đòi hỏi gì;  
(òn VBA cho chúng ta khả năng lập những hàm tùy thích miễn tuân thủ luật chơi về ngôn ngữ.   
Trở lại trường tạm nêu trên; Vì trường này chỉ dùng trong hậu trường, và cũng suy ra được từ ngày cần BC, nên độ dài 2 ký tự là đủ. (Sau này có dịp ta sẽ xét kỹ hơn). Hiện thời ta biết nếu dùng Advanced Filter theo trường đó, ta sẽ có số liệu trong nguyên 1 tháng;  
Các bạn thử dỡ trang tính ‘CSDL’ ra và thử nhiều lần đi, sẽ thấy tôi nói không sai. Và nếu cần bạn ghi thành macro luôn, khi đã thành thục.  
/(/hưng đã nói trên, số liệu nguyên tháng nhiều khi không cần thiết & sẽ dẫn đến thông tin lệch lạc. Vậy là tại vùng kết xuất kết quả tháng, ta phải thực thi nhiệm vụ xóa đi những dữ liệu sau ngày yêu cầu báo cáo!

**4./ Xóa đi số liệu các ngày sau của tháng**

Macro BaoCao dưới đây, tôi đã sửa lại, bằng cách thay vì những dòng lệnh dài tôi thay bằng cách gọi một macro khác (những macro mà tôi đã khẩn khoảng đề nghị các bạn thực hiện một cách riêng rẽ, như sau (Tên là do tôi áp đặt, các bạn nên xem lại):  
Macro xếp CSDL theo ngày (tăng dần) có tên Sub XepNgay()  
Macro lọc số liệu theo ngày có tên Sub LocNgayBC()  
Macro lọc số liệu theo tháng có tên là Sub LocThang()  
Macro xếp theo tháng có tên Sub XepThang(), Về nội dung macro này bạn tham khảo trong file đính kèm! Nó cũng giống như XepNgay & tiện cho ai đó cần xem xét, kiểm tra.   
Như vậy macro mà tôi có được sẽ trở thành như sau:

Code:

**Sub BaoCao()**

501 Application.ScreenUpdating = 0: Sheets("CSDL").Select

'Xếp theo Ngày

XepNgay

'Xóa số liệu cũ:

503 Range("L2:M125", "O2:Q125").Select: Selection.ClearContents

'Lọc số liệu ngày:

LocNgayBC

'Lọc số liệu tháng:

LocThang

'Xếp theo tháng:

XepThang

'Tính lũy kế đầu tháng:

507 Dim Dat As Date: Dim StrC As String

Dim Zj As Integer

509 Dat = Range("H2").Value

510 For Zj = 2 To 125

511 StrC = "O" & CStr(Zj): Range(StrC).Select

512 With Selection

513 If Not IsDate(.Value) Or .Value > Dat Then Exit For

514 End With

515 Next Zj

516 Range("O" & CStr(Zj) & ":Q125").ClearContents

Sheets("BCao").Select

518 Exit Sub

'Hiện các dòng dữ liệu:

Selection.Rows.Hidden = False

For Zj = 12 To 19

If Range("P" & CStr(Zj)).Value = 0 Then Rows(Zj).Hidden = True

Next Zj

**End Sub**

Câu lệnh đầu tiên tại 501 dùng để khỏi lắc lư màn hình khi ta Select từ sheets này sang sheets kia, hay từ đầu CSDL đến ngày cuối CSDL thậm chí đến dòng cuối của trang tính. Câu thứ 2 trong dòng ta đã biết;  
Hai câu lệnh trong dòng 503 sẽ chọn vùng gồm 2 khối & làm trống hoàn toàn các ô trong nó.  
/(ể từ câu lệnh 507 thực sự cần khảo sát kỹ; trong dòng đó có câu khai báo một biến Dat có kiểu dữ liệu ngày tháng; Kiểu dữ liệu này na ná với kiểu dữ liệi Long, vì thực ra ngày tháng là số nguyên kể từ 1/1/1900 được ấn định là 1, và cứ thế. . . cứ sau 1 ngày tăng 1 đơn vị; Biến này được gán dữ liệu chứa trong ô H2 (tất nhiên trong H2 phải chứa dữ liệu kiểu ngày hay tương tự.) thông qua câu lệnh 509;  
Dòng lệnh 510 & 515 ta cho qua?! Dòng 511, ở lệnh đầu là: đổi trị trong biến Zj sang dạng chuỗi và nối vô ký tự ‘O’; VD Zj đang là 35 thì biến StrC sẽ chứa chuỗi ‘O35’ (không có hai dấu nháy đơn). Dòng sau bảo với VBA chọn ô có địa chỉ như vậy làm ô hiện hành. Câu lệnh 512 & 514 ta qua luôn?! Câu 513 nói rằng nếu giá trị chứa trong ô hiện hành lớn hơn biến chứa ngày tháng (Dat) hay ô này không chứa giá trị ô ngày tháng thì thoát khỏi vòng lặp; Tại sao phải có hai ĐK (điều kiện)?; ĐK đầu áp dụng cho trường hợp ngày BC nhỏ hơn ngày sản xuất & đã nhập số liệu; ĐK sau để đúng với trường hợp ngày BC cũng là ngày nhập số liệu cuối cùng. Nếu thiếu một trong hai có lúc ta bị báo lỗi không ngờ.   
Tóm lại, vòng lặp 510 . . .515 sẽ lần lượt duyệt từ ‘O2’ cho đến hết ô 125 hoặc gặp một trong hai điều kiện nêu trên được thõa.  
Tuy ra khỏi vòng lặp rồi, nhưng VBA vẫn nhớ hoài giá trị Zj. Nhớ cho đến khi nó gặp các mệnh đề như Exit Sub, End Sub, hoặc ta gán cho nó giá trị mới . . .   
(ũng nhờ vậy mà ta viết được lệnh 516: Làm trống toàn bộ các ô trong vùng hình chữ nhật. Mà góc trên trái nhất của hình chự nhật là ô hiện hành cho đến góc phải thấp nhất của hình chữ nhật là Q125.   
Mà tại sao là 125 & cứ 125 mãi vậy?! Nó tùy thuộc vô chu trình sản xuất của phân xưởng II này; Khi ta là người ngoài lập cho nhân viên thống kê phân xưởng thì ta phải hỏi số tối đa trong tháng mà ca nào đó có thể sản xuất bao nhiêu sản phẩm.  
(/ậy sẽ nẫy sinh câu hỏi: Nếu sau này sản xuất phát triển, gây ra chuyện vượt số 125 dòng trong tháng thì làm cách nào?!  
Thực tế, các Lão luyện trong VBA khuyên chúng ta trong trường hợp này là khai báo 1 hằng số trong phần Declarations ở trên cùng của cửa sổ VBA, câu lệnh như sau:  
Const DongCuoi = 135  
Sau đó trong toàn bộ các macro trong đó có thể xài hằng số này; Một khi cần bảo trì hay nâng cấp, ta khỏi tìm trong các đoạn mã một cách khó khăn mà chỉ việc tăng trị cho hằng số mà thôi. Kể ra nói không sai rằng định luật bảo toàn cũng đúng cả trong trường hợp như vầy!  
Tại vùng Declarations này ta còn có thể khai báo nhiều thứ khác nhau, như  
Số Pi: Bản thân máy đã có hằng số này; nhưng nó rất dài, ta có thể cần số như 3,142 thôi:  
Const Pi = 3.142  
Hằng số còn có thể là chuỗi kí tự như   
Const Schu = “Chúc Mừng Đến Với VBA”  
Để sau đó ta lập macro như sau:

Code:

Sub Auto\_Open()

MsgBox Schu

End Sub

/(/ếu cũng chưa thỏa mãn theo kiểu phải khai báo hằng số thì có thể dùng vòng lặp Do. . . Loop thay cho vòng For . . . Next. Cú pháp để thực hiện công việc giống với công việc mà for . . . next làm như trên là vầy:  
‘Thay cho câu lệnh 510 bằng 3 câu lệnh 5100, 5101 & 5102:

Code:

5100 Zj =1

5101 Do

5102 Zj= 1 + Zj

511

512

513 ‘ Câu lệnh này If Not IsDate(.Value) Or .Value > Dat Then Exit For

‘ được thay bằng

5130 If Not IsDate(.Value) Or .Value > Dat Then Exit Do

514 End With

515 ‘ Next Zj được thay bằng

5150 Loop

*Bài tập của bài 5: Voọch cho kỹ càng file đính kèm*

**Bài 6 Macro thống kê số liệu.**

**1./ Thống kê sàn lượng các ca từ đầu tháng:**  
  
Thực ra TK (thống kê) này này đã tiềm ẩn, chỉ có điều ta chưa cho nó xuất đầu lộ diện ra mà thôi. Tại sheets("BCao") vùng từ P12-P20 có công thức tựa như sau

Trích:

|  |
| --- |
| =SUMIF(CSDL!P$2:P124,"B1?",CSDL!Q$2:Q124) |

(/ậy điều gì sẽ xuất hiện khi công thức trên được sửa lại thành:

Trích:

|  |
| --- |
| =SUMIF(CSDL!P$2:P124,"??A",CSDL!Q$2:Q124) |

và nhập vô ô nào đó (VD $D$4 trong sheets ('ThKe')). Công thức này cho ta SL (sản lượng) ca A từ đầu tháng. (Nếu bạn nào còn nghi ngờ nên xem lại thêm hàm SUMIF()!)   
  
/(/hư vậy ta chỉ việc copy xuống dưới 2 dòng nữa & chỉnh sửa lại, là ta có bảng SL các ca từ đầu tháng   
  
Trong Sheets("BCao") đã có SL từng ca trong ngày. Số liệu tìm được là do cộng dồn SL các sản phẩm sản xuất trong ngày. Nếu các bạn thích xài hàm SUMIF() thì cũng có công thức để tính SL của ca A như sau

Trích:

|  |
| --- |
| =SUMIF(CSDL!L$2:L124,"B?A",CSDL!M$2:M124) |

Ta có thể áp công thức này vô ô $C$4 của trang tính thống kê & chép xuống 2 ô còn lại, sửa ký tự 'A' thành 'B' & 'C' tương ứng sẽ có số liệu thống kê SL trong ngày theo ca;   
  
Chú ý: Trong bảng báo cáo, SL các sản phẩm của từng ca được tính theo hàm VLOOKUP(); Trường hợp gì sẽ sẩy ra khi người nhập liệu nhập 2 lần một số liệu: Hàm VLOOKUP() chỉ trả về 1 kết quả; nhưng hàm SUMIF() thì cộng cả hai record này. Như vậy bằng cách này ta có thể kiểm tra lại kết quả nhập liệu một cách gián tiếp!   
  
**2./ Thống kê sàn lượng từ đầu tháng theo ngày:**  
  
Thực ra ta chỉ cần một số thao tác tương tự như trên sẽ có số liệu những ngày sản xuất của phân xưởng II kia, thật vậy:   
  
Tại cột A, từ A10 đến A40 ta nhập các số tăng dần kể từ 1 & sau đó cho ẩn cột A đi!   
  
Tại B10 ta nhập hàm =DATE(YEAR(Nhap!B$2),MONTH(Nhap!B$2),A10)   
  
Tại C10 ta nhập công thức =SUMIF(CSDL!O$2:O125,B10,CSDL!Q$2:Q125)  
  
Sau đó tô chọn B10 & C10, chép xuống đến hàng 40.  
  
Trong công thức tại cột B vừa lập, ô Nhap!B$2 là ô được lấy làm ngày tháng cho báo cáo tại sheet("BCao"). (Xem thêm công thức tại tiêu đề báo cáo.)   
  
Như vậy ta đã thực hiện thống kê số liệu như đề mục đã nêu mà không phải dùng một câu lệnh nào hết! Hơn nữa vùng này luôn sẽ được cập nhật nếu hàng ngày ta cập nhật báo cáo bằng nút lệnh ' Báo cáo' tại Sheets("Nhap");  
  
/(/hưng số liệu không thỏa mãn chúng ta lắm, nhất là những ngày đầu tháng với quá nhiều số không trong dữ liệu;   
  
Giờ đây macro mới ra tay: làm ẩn các dòng chứa dữ liệu 0 trong vùng;  
  
Bạn hãy tự thu một macro Sub An1Dong() theo các bước sau:   
  
- Chọn dòng bất kỳ;  
  
- Vô menu Format -> Row -> Hide  
  
Và kết thúc việc thu; nhấn ATL+F8 để mở xem nội dung macro

Code:

**Sub An1Dong()**

Rows("40:40").Select: Range("B40").Activate

Selection.EntireRow.Hidden = True

**End Sub**

Để ẩn các dòng mà tại cột C của dòng đó có trị 0 ta phải dùng vòng lặp. Vòng lặp sẽ chọn lần lượt từng ô kể từ C40 -> C10; khi nào giá trị tại C(i) = 0 thì cho ẩn đi. Nội dung của nó như sau:

Code:

**Sub AnDong()**

Dim jZ As Integer: Dim StrC As String

Application.ScreenUpdating = False

601 Sheets("ThKe").Select: Rows("10:40").Select

602 Selection.EntireRow.Hidden = False

For jZ = 39 To 9 Step -1

StrC = "C" & CStr(jZ): Range(StrC).Select

With Selection

If .Offset(1, 0).Value = 0 Then

607 Rows(CStr(.Offset(1, 0).Row) & ":" & CStr(.Offset(1, 0).Row)).Hidden = True

End If

End With

Next jZ

**End Sub**

Ở đây chỉ có ba vấn đề tương đối mới cần giải nghĩa:  
  
Các câu lệnh 601:602 là chọn vùng gồm các hàng từ 10 đến 40 & cho hết ẩn. Tại sao phải làm vậy? Bỡi lẽ đề phòng ai trước đó đã cho ẩn một số dòng bằng macro này hay một macro khác siêu việt hơn.   
  
Tại sao vòng lặp lại bắt đầu từ ô 39 mà không là 40? Vì chúng ta cho ẩn không phải dòng hiện hành mà là dòng dưới dòng hiện hành 1 dòng. (Còn hỏi tiếp: Tại sao làm vậy? – Thích vậy, thế thôi!).   
  
Dòng 607 được diễn dịch là: cho ẩn dòng ngay dưới dòng hiện hành. (Điều kiện đã nêu trong dòng lệnh trên : Nếu dữ liệu trong ô C của nó bằng không.)   
  
/(/hưng macro này cần được cải tiến, vì còn chưa phải là thời đại 2K (hay thời @). Tại vì giống như đi bộ, thay vì phải chạy. . . Tôi xin giới thiệu một macro ẩn một lúc nhiều dòng trống, một khi chúng liên tục, như sau:

Code:

**Sub AnNDong()**

Application.ScreenUpdating = False

Sheets("ThKe").Select: Rows("10:40").Select

Selection.EntireRow.Hidden = False

Dim jZ As Integer, iBDau As Integer: Dim StrC As String

For jZ = 40 To 10 Step -1

StrC = "C" & CStr(jZ): Range(StrC).Select

With Selection

608 If .Value = 0 Then

609 If iBDau = 0 Then iBDau = .Row

610 Else

611 If iBDau > 0 Then

612 Rows(CStr(iBDau) & ":" & CStr(.Offset(1, 0).Row)).Hidden = True

613 iBDau = 0

614 End If: End If

End With

Next jZ

**End Sub**

Nếu là người mới vô nghề dịch thuật thì phải kè bên mình quyễn từ điển. Còn chúng ta mới vô nghề đọc câu lệnh cũng phải có chí ít cây bút & tờ giấy nháp. Chúng ta sẽ vẽ sơ đồ khối của nhóm lệnh này!   
  
Từ phía dưới tờ giấy ta kẻ 1 đoạn thẳng & ghi số 40; Sau đó ghi trên nó giữa đoạn thẳng là số 0; (chỉ những ngày cuối tháng của tháng có 31 ngày mới sản lượng mà thôi);   
  
Câu lệnh 609 yêu cầu gán số hàng vô iBDau nếu biến này đang chứa giá tri 0. (Cần nhớ rằng sau khi khai báo iBDau, lập tức nó sẽ có giá trị 0; Còn khi khởi tạo biến dạng chuỗi nó sẽ chứa chuỗi rỗng "" (khác với chuỗi trắng " ")). Ngoài ra ta còn phải hiểu thêm rằng nếu iBDau đã lớn hơn 0 thì gặp ô giá trị 0 kế tiếp biến này cũng không thay giá trị. (Coi như câu lệnh bảo ta rẽ trái)   
  
Câu lệnh 610 nói rằng: trị trong ô 'C(i)' <> 0 thì thực hiện các lệnh 611 đế 613, (Coi như câu lệnh bảo ta rẽ về đường bên phải gọi là nhánh (b)). Nhưng tại nhánh (b) này cũng lại gặp ngã 3; Hướng rẽ trái (ta cho là vậy), & là hướng chính, đó là: nếu biến iBDau >0 thì là hai việc sau:   
  
a) Làm ẩn đi các dòng từ dưới ngay dòng hiện hành cho tới dòng trùng với trị trong biến iBDau đang lưu giữ;   
  
b) Sau khi cho ẩn xong thì gán cho tôi iBDau trở về 0  
  
Còn hướng rẽ phải: iBDau =0 thì đi tiếp ( Có nghĩa là lại tiếp tục gặp dòng mà tại cột C của nó giá trị chứa trong nó <> 0)   
  
Chú ý: Câu lệnh 609 được phép viết trên cùng một dòng thay vì phải ghi 3 dòng; Hơn nữa ta có thể ghi câu lệnh như sau:   
  
If Bdem = False then Ixxx = 0 Else Ixxx = 1  
  
Điều này ít thấy, nhưng cũng nêu luôn ra đây để cùng ghi nhận: Câu lệnh 614 thực tế là 2 dòng lệnh, được ngăn cách bỡi ký tự ':'. Nhưng khi gặp hai vòng lặp For lồng vô nhau, ta có thể ghi như sau mà không bị phản đối:

Code:

*For Jz = 1 to 999*

*iDem = iDem +1*

*for Wj = 1 to 987*

*iDem = iDem -1*

*Next Wj, Jz*

Như các bạn thấy biến đếm vòng lặp trong ghi trước biến đếm vòng lặp ngoài; Nếu ngược lại thì hai vòng lặp này không còn lồng vô nhau nữa, mà đang cãi nhau!

3./ Xóa dòng trong sheets  
  
Ta thu macro xóa một dòng theo các bước như sau:  
  
Vô menu Tool ->Macro -> Record New Macro. . . Và đặt tên cho macro là Xoa1Dong  
  
Chọn dòng 45 & vô menu Edit -> Delete  
  
Và bấm nút kết thúc macro. Macro có nội dung như sau:

Code:

**Sub Xoa1Dong()**

Rows("45:45").Select

Selection.Delete Shift:=xlUp

**End Sub**

Để thấy macro này làm việc ra sao ta phải tô màu nền các ô từ A45 cho đến A50 bằng cách bấm vô trang tính, chọn dãy ô & chọn 1 trong các màu dòng cuối trong biểu tượng Fill Color trên thanh công cụ. Sau đó ta bấm vô CS (cửa sổ) Microsoft Visual Basic & thu nhỏ nó lại chỉ chiếm nữa phải màn hình; Đặt con trỏ trong các dòng của macro & nhấn phím F5 vài lần; Cứ sau mỗi lần nhấn, ta sẽ thấy các ô tô màu nền giảm đi một.   
  
Sau đó ta để con trỏ sau chữ Select & nhấn phím Delete để hai dòng lệnh trở thành 1 Và xóa chữ "Select" & "Selection." đi. Ta tiếp tục bấm {F5} để macro tiếp tục xóa các dòng có màu còn lại.   
  
(ông việc tiếp theo sẽ thử thay thế macro ẩn dòng bằng macro có chứa lệnh xóa dòng. Các bước như sau:   
  
Tô chọn dãy các ô từ A10 đến C40 & chép xuống A50 trở đi  
  
Tại đầu dòng lệnh 603 ta đánh dấu nháy để vô hiệu hóa đi; đến cuối dòng lệnh & bấm ENTER tạo dòng mới. Nhập vô đó câu lệnh:   
  
For jZ = 80 To 50 Step -1  
  
Tiếp theo cũng làm như vậy với câu lệnh 612 & nhập câu lệnh mới  
  
Rows(CStr(iBDau) & ":" & CStr(.Offset(1, 0).Row)).Delete Shift:=xlUp  
  
Thu nhỏ CS MVB còn nữa phải màn hình; bấm trỏ chuột lên macro AnNDong & nhấn phím {F5} để xem macro làm việc;   
  
Sau khi thử nghiệm thông thạo ta trả macro AnNDong về như cũ. Vì thực tế chúng ta cần là ẩn dòng, chứ không cần xóa dòng trong báo cáo   
  
**Bài tập của bài 6:**

Tại Sheets("Nhap") có ComboBox "Sản lượng". Hãy phải chuột vô đó, chọn dòng gần cuối Assign Macro. . . . Khi CS Assign Macro xuất hiện hãy chọn nút Edit bên phải CS để xem nội dung macro này. Bạn có hiểu nội dung không?

**Bài 7 Macro và đồ thị.**

***1.Dựng đồ thị dạng cột bằng VBA***  
  
Thật ra chúng ta sẽ thu Mc (macro) tạo lập một vài loại ĐT (đồ thị). Tại Sheets(“DoThi”) ta nhập vô cột A bắt đầu từ A2 các giá trị tùy chọn sau: 5, 4, 7, 9, 6, 7. Và A1 là chuỗi ‘SL’. Ta thực hiện các bước thu Mc tự động xây dựng ĐT dạng cột, như sau:  
  
. Vô menu Tool ->Macro-> Record New Macro & đặt tên cho Mc là DoThi1  
  
. Chọn vùng A2:A7 bấm vô biểu tượng Chart Wizard trên ToolBar. Sẽ xuất hiện bước 1 của Chart Wizard (Chart Type). Ta chấp nhận theo gợi ý của Excel bằng cách bấm Next. Chuyển nhanh sang bước 3 bằng nút lệnh Next một lần nữa;  
  
. Tại giai đoạn ba ta vô ngăn Gridlines & bỏ chọn nút Major Gridlines. Chuyển sang bước bốn bằng phím next & bấm nút lệnh Finish để kết thúc bước bốn của việc vẽ ĐT.  
  
. Bấm phải chuột vô vùng màu sám của ĐT & chọn hàng trên cùng có dòng Format Pilot Area. . .. Trong hộp thoại này ta thấy phía phải có hộp Area. Ta chọn màu trắng thay vì màu xám  
  
. Kết thúc thu Mc.  
  
Ta mở Mc ra trình tự sẽ sửa & chạy lại nó như sau:  
  
- Vô hiệu hóa các dòng lệnh sau

Code:

With ActiveChart.Axes(xlCategory)

.HasMajorGridlines = False

.HasMinorGridlines = False

End With

With Selection.Border

.ColorIndex = 16

.Weight = xlThin

.LineStyle = xlContinuous

End With

- Thu nhỏ CS (cửa sổ) VB (Microsoft Visual Basic) để nhìn thấy một nữa đồ thị; Thiết đặt trỏ chuột trên Mc vừa sửa & bấm phím {F5} cho Mc chạy. Nếu Excel không đưa ra lời phản đối nào thì ta có hai độ thị; bạn thử bỏ bớt một cái;  
  
/(hi đó Mc sẽ có nội dung như sau:

Code:

**Sub DoThi1()**

Range("A2:A7").Select

Charts.Add

ActiveChart.ChartType = xlColumnClustered

ActiveChart.SetSourceData Source:=Sheets("DoThi").Range("A2:A7"), PlotBy:= \_

xlColumns ActiveChart.Location Where:=xlLocationAsObject, Name:="DoThi"

With ActiveChart.Axes(xlValue)

.HasMajorGridlines = False

.HasMinorGridlines = False

End With

ActiveChart.PlotArea.Select

With Selection.Interior

.ColorIndex = 2

.PatternColorIndex = 1

.Pattern = xlSolid

End With

**End Sub**

***2.Dựng đồ thị dạng đoạn thẳng & làm cho Mc nhận thông số ta trao cho***  
  
Nếu tại dòng thứ ba ta vô hiêu hóa nó bằng dấu nháy đầu dòng; sau đó tạo dòng & nhập vô dòng mới này câu lệnh sau:  
ActiveChart.ChartType = xlLineMarkers  
Cho chạy lại Mc với dòng lệnh mới này, bạn sẽ thu được một ĐT dạng đoạn thẳng gấp khúc;  
(Dòng lệnh này sẽ có trong Mc nếu thu như Mc DoThi2 như ở phần đầu của bài, nhưng ở bước 1 ta chọn loại đồ thị đoạn thẳng)  
Như vậy ta có thể nhận hai loại ĐT trên một Mc; Nhưng còn một điều bất tiện là mỗi lần cứ phải đổi dòng lệnh hay sao?! Có cách nào để Mc tự đi đúng hướng ta cần không?.   
Có & đó là cách chúng ta trao cho Mc một biến (một tham số) để nó thực hiện; Biến đó tùy lúc nhận một trong hai số liệu quy ước của excel mà nó hiểu được, cụ thể ở đây là xlColumnClustered & xlLineMarkers; (các biến này thuộc kiểu dữ liệu Integer)  
Ta thay Mc đó như sau

Code:

**Sub DoThi1(LoaiDT As Integer)**

Range("A2:A7").Select

Charts.Add

‘ ActiveChart.ChartType = xlColumnClustered

ActiveChart.ChartType = LoaiDT ‘xlLineMarkers

. . . . . . . .

**End Sub**

Sau khi sửa & lưu Mc này lại, trở về bảng tính & bấm tổ hợp phím ALT+{F8} ta không còn thấy tên Mc DiThi1 được liệt kê trong danh sách. Nhưng nó không mất đâu các bạn! Nếu bạn đã lập một Mc nào khác thì bấm vô đó; bằng không bạn bấm ALT+{F11} thì CS VB sẽ xuất hiện để bạn xem xét & thao tác với nó nếu cần!  
  
Cách gọi Mc DoThi1 này khi đến mục 4 của bài sẽ đề cập đến. Còn tại thời điểm này, muốn kiểm tra sự sửa chữa của bạn đã đáng yên tâm chưa, ta vô CS VB như vừa nêu trên. Sau đó thu nhỏ CS VB này còn chiếm nữa màn hình. Tiếp theo vô menu View -> Immediate Window. Ta sẽ thấy CS mới xuất hiện & khiêm tốn nằm bên dưới CS VB. Ta ấn chuột vô thanh tiêu đề CS Immediate này & kéo cho nằm trung tâm của CS VB  
  
Bạn bấm trỏ chuột lên CS Immediate & nhập dòng lệnh DoThi1 xlColumnClustered Kết thúc bằng phím ENTER để Mc được thực hiện với tham số xlColumnClustered. Nếu kết quả mỹ mãn, bạn ghi tham số thứ hai vô & kiểm chứng lần nữa để hoàn toàn yên tâm.  
  
Sau đây ta lập thêm một Mc biểu đồ hình bánh đơn giản nữa. Vì sao tôi nói đơn giản, vì chúng ta chưa tinh chỉnh cho các ĐT dễ nhìn hơn một tý tẹo nào cả!  
  
***3.Dựng đồ thị dạng hình bánh đơn giản & có miếng tách rời***  
  
Ta tạo Mc có tên DoThiB bằng cách vô menu Tool ->Macro->Record New Macro. . . như từng quen thuộc. Thực hiện các bước tiếp theo cũng rất quen với chúng ta, đó là:  
  
- Quét chọn vùng dữ liệu từ A2 đến A7; bấm vô biểu tượng tạo ĐT trên thanh công cụ;  
  
- Tại bước 1 của 4 bước tạo ĐT, chúng ta chọn Pie trong CS Chart Type; Tại CS Chart Sub-Type bên phải xuất hiên 6 loại bánh; ta chọn loại giữa hàng trên;  
  
- Chấp nhận mặc định của các bước hai & ba & ấn ngay phím Finish ở bước 4;  
  
- Tại ĐT hình bánh mới xuất hiện, ta kích hoạt các bánh & sau đó chọn một bánh (khi được chọn, trên viền viên phân đó xuất hiện 6 nút đánh dấu; ta ấn chuột vô viên phân đó & kéo rê ra xa tâm của bánh một chút  
  
- Ta kết thúc thu Mc tại đây;  
  
/(/ếu Mc của bạn giống như dưới đây là tốt rồi; Tại hai dòng cuối của Mc đề cập nói lên viên phân thứ mấy trong các viên phân được chọn & kéo ra khỏi các tập hợp của chúng

Code:

**Sub DoThiB()**

Range("A2:A7").Select

Charts.Add

ActiveChart.ChartType = xl3DPie

ActiveChart.SetSourceData Source:=Sheets("DoThi").Range("A2:A7"), PlotBy:= \_

xlColumns

ActiveChart.Location Where:=xlLocationAsObject, Name:="DoThi"

ActiveChart.HasTitle = False

ActiveChart.SeriesCollection(1).Select

ActiveChart.SeriesCollection(1).Points(4).Select

Selection.Explosion = 30

**End Sub**

*Ghi chú:* Vì ta ghi Mc bằng những hành động thật đơn giản nên các bạn tự suy luận & dịch ngôn ngữ VBA hai Mc này!  
Nếu bạn nào không tự tạo được Mc cho mình thì cách cuối cùng là chép hai Mc này vô CS VB của bạn để xài tạm;  
Trong thực tế, việc chỉnh dáng ĐT không lúc nào là không thực hiện; nhưng ở đây chính là chúng ta quan tâm đến việc tạo ĐT bằng VB. Còn sau khi đã thành thục thì nâng độ khó của Mc là chuyện của bạn. Hơn nữa sau khi chạy Mc với những số liệu khác nhau, bạn phải chuột vô phần ĐT cần chỉnh sửa & hoàn thiện chúng cũng không muộn.

***3.Dùng OptionButton gọi thực hiện một Mc***

/(/ói trước là trong excel có hai loại OB (Option Button); Loại đầu tiên ta sẽ thấy ngay sau đây; loại thứ hai ở trong thanh công cụ Forms. Muốn nhanh chóng thấy thanh công cụ này ta phải chuột lên vùng trống của ToolBar & chọn dòng Forms.  
(òn muốn làm việc với loại đầu tiên ta cũng phải chuột vô vùng như trên; Sau đó chọn dòng Control Toolbox. Khi đó xuất hiện trên màn hình thanh công cụ Control Toolbox gồm thường 15 nút. Nút chúng ta quan tâm là nút khi ta để chuột vô nó một thời gian đủ lâu, sẽ hiện lên dòng Option Button.  
  
Ta nhấp chuột lên nút & đến vùng nào đó còn trống mà bạn muốn đặt nút này lên. Ấn trái chuột & kéo vẽ hình chữ nhật chiếm trọn 4 ô (VD B9:C10); bấm lên chỗ trống trên trang tính. Sau đó bạn phải chuột lên nút lệnh để xem menu tắt của nó. Trên nó có dòng View code ta ấn vô đó. Lập tức CS VB mở ra & có dọn sẵn cho ta bữa diểm tâm gồm:

Code:

**Private Sub OptionButton1\_Click()**

**End Sub**

Trích:

|  |
| --- |
| Dòng lệnh ta thêm vô giữa hai dòng này: msgbox “1”,,”OK” |

Xong ta đóng CS VB bằng tổ hợp ATL+Q  
  
Thực hiện lặp lại với 1 OB nữa; nhưng lần này với dòng lệnh msgbox “2”, , “OK”  
  
Tiếp tục phát huy chiên quả với OB thứ ba, chỉ khác dòng lệnh giờ là dothib (đó là tên Mc tạo ĐT dạnh bánh mà ta đã thu. Nếu không sai thì VBA sẽ sửa tên của MC ta vừa gỏ trở thành đúng của nó (đó là DoThiB)  
  
Trở về trang tính, sau một vài phút (suy) nghỉ, ta tiếp tục phải chuột vô cac OB & chọn format Control; vô ngăn Properties bỏ chọn nút Print Object, nhưng lại chọn nút Don’t move or size with Cells. (Việc này nên thực hiện cho cả ba OB mới có í nghĩa)  
  
Lại phải chuột vô OB, chọn dòng OptionButton Object => Edit; Con trỏ nháy lên các chữ mặc định của Excel; Ta sửa chúng lần lượt là: Đồ thị dạng cột; Đồ thị đoạn thẳng & Đồ thị dạng bánh.  
  
Kết thúc công việc xem ra mới mẽ & kó ưa này bằng cách thoát về windows để lưu mọi thao tác của ta lại;  
  
Sau đó lại mở trang tính Dothi đó ra trong excel; chúng ta thử chọn lên lần lượt từng nút OB xem điều gì sảy ra?!; Phải là vầy:  
  
Ấn lên nút (1) sẽ hiện lênh hộp thoại Dòng tiêu đề có chữ ‘OK’, dưới nó là dòng ‘1’ & nút ‘OK’; Bạn phải bấm vô nó mới làm việc khác được!  
  
Nút thư hai cũng tương tự; Nút thứ ba bị chọn sẽ xuất hiện ĐT hình bánh!

***4. Một Mc truyền thông số cho một Mc***  
  
Ta biết rằng Mc DoThi1 cần được nhận lần lượt hai biến khác nhau để vẽ lên hai dạng đồ thị khác nhau; Ta đã có 2 OB; như vậy việc còn lại sẽ như con thỏ ăn cỏ mà thôi:  
  
Trước tiên mà cần biết tìm nhanh 3 cái Mc VB tạo ra chúng nằm đâu?  
  
Ta phài chuột vô ô DoThi của thanh liệt kê trang tính (góc trái dưới màn hình;  
  
Ta vô thăm những gì có trong dòng View Code; Sẽ đúng nếu ta lại thấy ba cái anh chàng mà ta & VB đã cùng nhau nặn ra hình hài  
  
Ta sửa hai anh đầu thôi, để có nội dung sau:

Code:

**Private Sub OptionButton1\_Click()**

DoThi1 xlColumnClustered

**End Sub**

**Private Sub OptionButton2\_Click()**

DoThi1 (xlLineMarkers)

**End Sub**

Chú ý:   
  
· Sau khi lưu & mở lại sheets(“DoThi”) ta không thể phải chuột vô các OB để sửa những sai lệch được nữa; (VD chuyển từ tên gọi ‘Đồ thị bánh’ sang ‘Đồ thị dạng bánh’); Chỉ có cách là cho hiện thanh công cụ Control Toolbox; Bấm vô nút có hình ê ke có Tool Tip là Design Mode, sau đó chọn OB cần hiệu chỉnh,. Sau khi hiệu chỉng xong ta lại bấm biểu tượng thước ê ke một lần nữa, lúc này phần đầu của Tool Tip có thêm chữ Exit   
  
· Thực ra hai cách truyền tham số như trên cùng được VB chấp nhận, ta nên chọn một cách nào thấy thuận tiện!   
  
Đến thời điểm này sẽ phát sinh các câu hỏi:  
  
*1.* Nếu nhu cầu truyền nhiều hơn một tham số thì sao?  
  
*2.* Chữ Private kia trước chữ Sub có nghĩa gì?  
  
*3.* Có cách nào khi Mc gọi lệnh thực hành Mc DoThi1 mà quên (hay vì lí do gì đó không tìm ra thông số đề) truyền tham số thì Mc Dothi1 cứ lấy một tham số nào đó & chạy hay không?  
  
Những vấn đề này sẽ đề cập tới trong những lần sau, nếu có dịp. Nhưng các bạn có thể tự tìm hiểu ở đâu đó trước xem sao?

**. .Bài 8 Macro lập PivotTable.**

***1. Tạo dựng một PivotTable nhờ macro***

Đầu tiên chúng ta cần tạo ra CSDL (cơ sở dữ liệu) như trong file Bai8.XLS đính kèm. CSDL trong sheets(“PiVot”) có 7 trường: Ngay, TenHang, NhaCungCap, SoLuong, Tinh, DonGia, TTien để quản lý hàng xuất nhập của các đơn vị thành viên ở Tp HCM & ba tỉnh lân cận.  
Ta thu Mc (macro) cũng như các lần trước. Nhưng với những ai ít thao tác tạo PivotTable thì phải nhuần nhuyễn trước các bước sau đây:  
Tại trang tính ‘PiVot’, ta vô menu Tool ->Macro -> Record New Macro. . . . Tại CS (cửa sổ) vừa xuất hiện ta đặt tên Mc là PiVotTable & bấm OK để về trang tính & thu tiếp các bước sau:  
· Ta bấm vô ô A11 (là ô có dữ liệu của CSDL) & ấn tổ hợp CTRL+\* (phím dấu sao). Khi đó toàn bộ khối dữ liệu được chọn.  
· Vô menu Data -> PivotTable and PivotChart Wizard – Step 1 of 3. trong CS sẽ có hai dòng mặc định được chọn, đó là:   
Microsoft office Excel list or database  
Pivot Table  
Ta bấm vô phím Next để qua giai đoạn 2; nếu toàn CSDL đã được liệt kê ta chọn bấm tiếp nút Next để qua giai đoạn 3.   
· Tại đây ta ấn vô nút Layout (trái nhất của CS). Màn hình thiết kế trang PivotTable xuất hiện; Ta kéo trường NCC vô vùng PAGE;  
Tương tự trường Tinh được kéo vô vùng COLUMN; kéo trường THg vô vùng ROW và cuối cùng là trường TTien vô vùng DATA.   
Nhấp đúp lên chính trường này trong vùng vừa thả để vô CS PivotTable Field; chấp nhận các thông số mặc định là tính tổng; nhấp tiếp vô nút Number. . . bên phía phải CS và chọn định dạng cho tổng thành tiền có 1 chữ số sau dấu phảy & ta cần phân cách phần ngàn. Xong bấm hai lần hai nút OK kết thúc phần tạo lập PivotTable;  
· Trên màn hình hiện ra bảng tổng hợp theo ta yêu cầu; Bên cạnh trường THg có hình mũi tên xuống rất đậm; ta bấm vô để thấy liệt kê danh sách mã hàng; ta bỏ chọn một mặt hàng có mã là RDE & bấm OK trở về trang tính; khi đó mất một dòng trên bảng biểu tổng hợp.  
· Kết thúc ghi Mc  
Mc của chúng ta thu có những dòng lệnh sau:

Code:

**Sub Pivot\_Table()**

Range("A11").Select ‘Ta khởi động Mc tại Sheets(“PiVot”)

802 Selection.CurrentRegion.Select ‘ Chọn toàn vùng CSDL

‘Với Mặcđịnh version 10, một PivotTable đích có tên “PivotTable1” từ dữ liệu nguồn

‘tại Sheets(“PiVot”) bắt đầu từ ô trái trên là dòng 10 cột 1 cho đến ô phải dưới dòng 50 cột 7

‘ được tạo thêm:

803 ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:= \_

"Pivot!R10C1:R50C7").CreatePivotTable TableDestination:="", TableName:= \_

"PivotTable1", DefaultVersion:=xlPivotTableVersion10

‘Lấy ô (3,1) của trang tính vừa lập làm nới bắt đầu dựng Pivottable:

804 ActiveSheet.PivotTableWizard TableDestination:=ActiveSheet.Cells(3, 1)

ActiveSheet.Cells(3, 1).Select ‘’(Bạn hiểu rồi!)

‘Tổng hợp theo NCC, với các danh sách cột trong trường [Tinh], các hàng là trường [THg]

‘vô trang tính có tên ‘PivotTable1’ (Trang này do VB gán cho):

806 ActiveSheet.PivotTables("PivotTable1").AddFields RowFields:="THg", \_

ColumnFields:="Tinh", PageFields:="NCC"

‘ Với trường [TTien] trong ‘PivotTable1, tiến hành định dạng dịnh dạng dữ liệu kểu số

‘ với 1 ký số ở phần thập phân:

With ActiveSheet.PivotTables("PivotTable1").PivotFields("TTien")

808 .Orientation = xlDataField: .NumberFormat = "#,##0.0"

End With

‘ Phần câu lệnh làm ẩn mặt hàng có mã RDE

With ActiveSheet.PivotTables("PivotTable1").PivotFields("THg")

810 .PivotItems("RDE").Visible = False

End With

**End Sub**

Thường những câu lệnh tạo PivotTable, Advanced Filter,. . . đều dài. Những câu như vậy có khi dễ hiểu hơn những câu ngắn, & không đến nổi làm ta không thể hiểu.

***2. Xóa một PivotTable nhờ macro***

Để xóa 1 trang tính (VD như trang PivotTable) một cách tự động, ta phải lợi dụng tính tự động gán tên trang tính khi Excel khởi tạo; Muốn vậy ta đổi tên các trang tính có tên mặc định đi, để tránh bị xóa nhằm.  
Sau đó ta chạy Mc PiVotTable để nó tạo ra 1 bảng tổng hợp;  
Thực hiện thu Mc để xóa bảng vừa tạo có tên XoaTrang. Nó có hai câu lệnh như sau

Code:

**Sub XoaTrang()**

Sheets("Sheet1").Select: ActiveWindow.SelectedSheets.Delete

**End Sub**

Mc này còn vài khiếm khuyết: Nó chỉ biết xóa trang tính có tên ‘Sheet1’ mà thôi; Không những thế nó còn hay hỏi lôi thôi! Để thấy điều này bạn thử cho chạy Mc PivotTable & sau đó xóa thử trang tính vừa tạo bằng Mc sẽ biết ngay!  
Chúng ta sẽ tác động đến Mc này để có năng lực xóa 9 trang tính nếu có trên workbook của ta & sẽ cắm đầu làm thôi, khỏi hỏi nữa! Nội dung Mc sẽ như sau:

Code:

**Sub ClearTable()**

On Error GoTo LoiMacro

Dim iJ as Integer: Dim StrC As String

Application.DisplayAlerts = False

For iJ = 1 To 9

StrC = "Sheet" & CStr(iJ): Sheets(StrC).Select

Worksheets(StrC).Delete

Next iJ

errMacro: Application.DisplayAlerts = True

Err = 0: Exit Sub ‘ ===\*===\*===

LoiMacro:

If Err = 9 Then Resume errMacro

If iJ < 9 Then

Resume Next

Else

MsgBox "Ban Hay Tu Xoa PivotTable Vua Tao!": Resume errMacro

End If

**End Sub**

Trong Mc này nhiệm vụ chủ yếu là: nếu gặp các trang tính có tên Sheet(i) mà i=1-9 thì xóa đi, Nội dung của nó được thể hiện trong vòng lặp For . . .. . Next.  
Để tránh việc hỏi lại, mà cứ xóa hết ta đã dùng câu lệnh:  
Application.DisplayAlerts = False  
Phía sau câu lệnh này, VB sẽ không hỏi chúng ta nữa cho đến khi gặp lại câu lệnh  
Application.DisplayAlerts = True  
Để tránh lỗi, trong Mc có bộ phận bãy & xử lý lỗi & dịch câu lệnh đầu tiên trong Mc là:   
LoiMacro là nơi đến nếu gặp lỗi   
Vậy cái nơi mà phải đến này làm những việc gì?  
a.) Nếu bạn bị lỗi có mã lỗi là 9 (Err = 9) thì đến địa chỉ errMacro để giải quyết;   
(Thực ra đến đây để rữa kiếm & nghỉ đó thôi: )  
b.) Nếu mã lỗi <> 9 thì: Sẽ thực thi một trong hai trường hợp sau:  
o Nếu iJ < 9 thì thực hiện các câu lệnh tiếp sau  
o Nếu iJ =9 thì đưa ra lời khuyên: Hãy tự xóa PivotTable vừa tạo & tiếp như a.)  
Trường hợp gì sẽ diễn ra nếu tôi có ba trang tính với các tên: ‘Sheet1’, ‘Sheet2’ & ‘Sheet4’? Khi chạy Mc ClearTable nó sẽ chừa lại trang tính cuối, để ta làm kỹ niệm!  
Để VB giải thích cho rõ thêm về lỗi có mang mã số 9 (cũng như các mã khác nữa) ta cho hiện CS Immediate như bài trước đã nêu. Gõ vô đó dòng lệnh: ? Error$(9) & ấn ENTER để biết thêm.

***3. Người dùng tự chọn lựa các trường***

/(/ếu ta đi qua ba bước tạo PivotTable cũng khá mất thời gian, nhất là chỉ cần khảo sát những chỉ tiêu một cách đơn giản nhưng lẹ làng.  
Tiếp theo chúng ta sẽ kết hợp Mc & sự hỗ trợ của ComboBox & listBox trong thanh công cụ Forms thì tăng tốc đáng kể việc tạo PivotTable;  
Để được vậy, trước tiên ta tạo Sheet có tên ‘CrTab’ theo file đính kèm. Trên nó ta gắn 1 LB (listbox) (tại đầu cột H) & dưới nó là một CB (ComboBox);. Gắn sẵn thêm một nút lệnh để chạy Mc tạo PT (PivotTable) (Nút đang có nhãn ‘New Pivot’).  
Khi trên LB hiện từ Row, nếu chọn trường nào đó trong CB thì Mc ẩn trong CB sẽ gán tên trường đó vô ô B3 (để Mc tạo PT lấy làm hàng). Tương tự nếu LB có từ Column thì sẽ gán đến C2 tên trường dùng để làm cột cho PT & sẽ là DATA nếu tại LB là Data.  
Nội dung hai Mc đó như sau:

Code:

**Sub ChonTruong()**

Dim Truong As Integer, SRng As String

Truong = Range("E1").Value ‘---->’E1’ là ô nối kết với LB

Select Case Truong

Case 1

SRng = "B3"

Case 2: SRng = "C2"

Case 3: SRng = "C3"

Case Else: Exit Sub

End Select

Range(SRng).Value = Application.VLookup(Range("E3").Value, Range("F1:G8"), 2)

**End Sub**

**Sub AddPivotTable()**

On Error Resume Next ‘ Câu lệnh này đã được đề cập đến

Dim sRField As String, sCField As String, sDField As String

Dim iRange As Range ‘<= Khai báo biến có kiểu dữ liệu là range

Application.ScreenUpdating = 0

sRField = Range("B3").Value ‘ß Thực hiện các phép gán giá trị cho các biến:

sCField = Range("C2").Value: sDField = Range("C3").Value

Sheets("PiVot").Select: Range("B12").Select

Selection.CurrentRegion.Select ‘ <= Chọn vùng

Set iRange = Selection ‘ <= Vùng chọn đem gán vô biến iRange (đã khai báo)

830 ActiveSheet.PivotTableWizard SourceType:=xlDatabase, SourceData:=iRange, \_ TableName:="PivotTable6"

With ActiveSheet.PivotTables("PivotTable6")

832 .AddFields RowFields:=sRField, ColumnFields:=sCField, PageFields:=sDField

833 .AddDataField ActiveSheet.PivotTables("PivotTable6").PivotFields(sDField), "Sum of " \_

& sDField, xlSum

End With

Application.ScreenUpdating = True

**End Sub**

Chú ý: Câu lệnh cuối cùng trong Mc trên được dịch là: Kết quả của hàm VLOOKUP() với các tham số tuơng ứng trong ứng dụng của Excel đem gán làm dữ liệu của vùng có địa chỉ ứng với SRng   
So với lệnh tạo PT mà ta ghi thông qua bộ thu Mc thì ở đây (dòng lệnh 830) khởi tạo PT bằng một cách tương đối khác. Dịch dòng lệnh đó là: PivotTable6 sẽ là tên, vùng dữ liệu chứa trong biến irange làm nguồn, khởi tạo 1 PivotTale thuộc loại cơ sở dữ liệu. Rồi sau đó mới định ra các thuộc tính của PT đó như: (832) Lấy dữ liệu trong biến sDField làm PageField, lấy dữ liệu trong biến sCField làm các cột; lấy dữ liệu trong biến sRField làm các hàng;  
(833): Tình tổng là tổng chứa trong biến sDField, tổng hợp dữ liệu chứa trong biến sDField và đưa vô làm trường dữ liệu trong PT.

**Bài 9 Cửa sổ Microsoft Visual Basic.**

***1. Cửa sổ VB (Microsoft Visual Basic) của sheets***

Cho đến trước bài này, khi muốn xem xét nội dung một Mc (macro) nào đó ta thường vô menu Tool->Macro ->Macros. . . ( hay cùng lúc 2 phím nóng **ALT+{F8}** để vô CS (cửa sổ) Mc.  
Nhưng có lúc cũng đã đề cập đến cách nhanh để vô VB riêng rẽ của một sheet cụ thể nào đó bằng cách phải chuột vô tên sheet đó trên thanh liệt kê tên sheets. Đề mục này chúng ta sẽ nghiên cứu sâu hơn về các thành phần sẵn có trong CS VB của một sheet này.  
Hãy mở lại bảng tính của bạn về bài đồ thị (bài 7); trong đó có các Mc dạng sau:

Code:

**Private Sub OptionButtonN\_Click()**

DoThi1 [ThongSo]

**End Sub**

(Mà ở đây N=1-3); Vấn đề quan tâm của chúng ta lúc này không phả là các Mc này, mà là CS VB hiện hành; Bạn kiểm lại xem có phải nó như dưới đây không?  
Thanh tiêu đề CS có ghi Microsoft Visual Basic – [WordBookName] – [SheetN (Code)]; Dưới đó là thanh menu & thanh công cụ; Dưới thanh công cụ có hai CS; CS bên trái gọi là Object; Cái kia: Procedure. Ta chủ tâm vô các CS này.  
/(/ếu bấm vô mũi tên hướng xuống trong CS trái ta thấy liệt kê ngoài tên 3 Mc trên còn có hai dòng: (General) & WorkSheet   
a./ Nếu chọn dòng trên cùng thì CS bên phải xuất hiện dòng (Declarations)  
Tại vùng này chúng ta có thể khai báo các hằng số cũng như các biến dùng chung trong các Mc bên dưới. Tại đây ta có thể khai báo các tùy chọn như: **Option Explicit**  
Tùy chọn này bắt buộc các biến phải được khai báo trước khi dùng; Điều này rất nên làm, nhất là những người mới vỏ vẽ về VBA.  
Để áp đặt mọi biến phải luôn được khai báo trong excel, tại CS VB ta vô menu Tool -> Option. Trong ngăn Editor của hộp thoại Option ta đánh dấu chọn vô hàng thứ hai: Require Variable Declaration & ấn OK để thiết đặt chúng kể từ thời điểm này. Khi thiết đặt như vậy, toàn bộ CS VBA của các modules, cũng như các sheets & workbook sẽ có dòng Option Explicit  
b./ Nếu chọn dòng cuối cùng thì bên phải sẽ có 9 đề mục hiện lên. Ta sẽ thử biết các loại hình nghệ thuật mới mẻ này:  
Nếu ta ân chuột lên dòng đầu; lập tức tại CS VB sẽ có 2 dòng cole xuất hiện

Code:

**Private Sub Worksheet\_Activate()**

**End Sub**

Để tiết kiệm thời gian ta sẽ nhập vô hết 6 Mc này câu lệnh có cùng dạng như sau  
Msgbox “[Tương ứng]” vô giữa 2 dòng lệnh trên  
Như trường hợp (1) trên sẽ là MsgBox “Activate”; Trường hợp (3) – “Right Click”;   
Áp dụng như vậy cho các dòng tiếp theo (trừ dòng 7 & 8)  
Như vậy dòng (9) sẽ là MsgBox “SelectionChange. Lưu & thoát về trang tính bằng menu của VB hay bằng phím ALT+Q.   
Ta thử bấm vô một ô trống bất kỳ, sẽ thấy hộp thoại tương ứng xuất hiện  
Tiếp theo ta nhập vố đó 1 giá trị 9 & bấm ENTER sẽ có 2 hộp thoại hiện lên mà cái trước ghi là ‘Change’;  
Tiếp nữa, trong hộp dưới số 9 ta nhập vô công thức = [9] + 13 & ENTER để kiểm chứng thứ tự của 3 hộp thoại; Sau đó ta phải chuột vô ô chứa công thức vừa lập xem sao?!  
Có nghĩa là đến giờ bạn gặp không ít phiền toái khi phải di chuyển trong trang tính, đi đâu cũng bị hỏi ‘giấy tờ’! Để bớt đi nổi phiền muộn ta vô hiệu hóa Mc có chứa hộp thoại SelectionChange đi.  
Bạn tìm các cách để hiện hết 6 hộp thoại & tự rút ra kết luận cho mình;  
Sau khi bạn đã nhuyễn với 6 hộp thoại ta vô hiệu hóa chúng đi vì chỉ gây phiền phúc.   
Xin giới thiệu các bạn một ứng dụng kỳ diệu của vấn đề này trong công việc nhập liệu của chúng ta. Tôi có bảng tính lưu giữ CSDL (cơ sở dữ liệu) gồm các trường [TT], [Ngay], [MaHg], [SoLg], [Dgia], [Ttien]. . . .  
Tôi muốn chỉ cần nhập mã hàng thì VB sẽ tự nhập cho tôi dữ liệu trường [Ngay] (ghi ngày tháng năm nhập/xuất hàng). Mc có nội dung như sau:

Code:

**Private Sub Worksheet\_Change(ByVal Target As Range)**

'Hàm Sử Dụng Tốt Trong Trường Hợp Nhập/Xuất Hàng Hóa Theo Ngày/Tháng

901 If Not Intersect(Target, Range("C:C")) Is Nothing Then

902 If Not IsEmpty(Target) Then

903 Target.Offset(0, -1).Value = Date -1

Else

905 Target.Offset(0, 1).Value = Empty

End If: End If

**End Sub**

Ở đây Target ta hiểu là một biến, nhưng không phải của ta, vì ta chưa khai báo bao giờ. Mà phải hiểu là của VB; & những gì của VB thì nên trách xa, nếu không sợ bị đòn! Biến này sẽ đại diện cho ô mà ta đang muốn thay đổi giá trị nó chứa;

Câu lệnh 901 trong Mc tác giả đã dùng phương thức Intersect ( Range1, Range2[,. . . ]). Đến thời điểm này ta chỉ cần hiểu nó như sau: Phương thức kiểm tra xem vùng ô Range1 có giao ( gặp, cắt, thuộc ,bao hàm . . .) với vùng ô Range2 hay không?  
Câu lệnh 901 được dịch là: nếu không là không giao giữa cột C với ô đang bị thay đổi thì (thực hiện lệnh 901 – 905)   
(Mệnh đề phủ định của phủ định chứ không hề đánh máy nhằm đâu!)  
902: Nếu ô bị thay đổi này không rỗng (vừa nhập mã hàng mới) thì (thực hiện lệnh 903)  
903: Dữ liệu ngày tháng hôm qua gán vô ô bên trái liền kề với ô đang bị thay đổi  
Như vậy khi ô đang thay đổi có nhập giá trị & nó thuộc cột ‘C’ thì gán vô ô cột B liền kề giá trị ngày tháng. Như vậy VB đã thay ta nhập giá trị ngày tháng ta cần;  
Chú ý: Điều gì sẽ diễn ra khi ta thêm vô cuối dòng 903 lệnh sau: (Có dấu ‘:’)  
:

Trích:

|  |
| --- |
| Target.Offset(0, -2).Value = 1 + Target.Offset(-1, -2).Value |

**2. Cửa sổ Project – VBA Project**

/(hi đang trong CS VB, ta vô menu View & chọn tiếp Project Explorer (CTRL+R) ta sẽ thấy xuất hiện bên trái một CS Project – VBAProject, mà trong nó hiện sơ đồ nhánh cây liệt kê các đối tượng của workbook gồm 2 dòng chính:

**Code:**

**Microsoft Excel Objects**

**Modules**

Trong hàng đầu liệt kê tất cả những worksheets, nhưng tên chủ yếu là của Excel gán cho (Còn tên ta gán là thứ yếu nên để trong dấu ngoặt). Như ta nhấp đúp vô tên nào đó, thì bên CS chính sẽ hiện lên các Sub của sheet đó.

Nếu chúng ta bấm đúp vô dòng Modules thì sẽ hiện liệt kê những modules đã [b]có trong workbook (VD module1, Module2. . . . .)  
Ta bấm vô hàng ThisWorkbook ( phía trên dòng Modules) Và đóng CS Project bằng nút Close góc phải phía trên của nó. CS này cấu trúc giống như ta đã từng gặp (của sheets). Nhưng khi ta chọn Workbook bên CS Object, thì bên Procedure xuất hiện rất nhiều phương thức lựa chọn. Ta nhập dòng lệnh sau vô phương thức Open:

Code:

**Private Sub Workbook\_Open()**

MsgBox "Open WB"

**End Sub**

Sau đó ta thoát về Excel & lại mở lại WorkBook này để xem làm việc của Mc.  
Chú ý: Một cách nhanh nhất để đến CS chứa module của WorkBook là: Bên trái hơn của menu File (CS Microsoft Excel) có biểu tượng excel; Nếu ta phải chuột vô nó, dòng cuối của menu đổ xuống sẽ là View Code; Ta bấm chọn dòng này ta sẽ ở ngay CS VB của workbook;  
Ta thay dòng lệnh Msgbox “Open WB” nêu trên bằng dòng lệnh sau:

Code:

Application.OnTime Now + TimeValue("00:00:09"), "MyMacro"

Diễn dịch câu lệnh này như sau: Chạy Macro có tên MyMacro sau 9 giây nữa.  
Tuy nhiên đến thời điểm này ta chưa viết Mc MyMacro. Hơn nữa Mc này nên có tại CS VB chuẩn. Từ CS module của WorkBook ta về CS module chuẩn theo 1 trong 2 cách:  
- Đóng CS module này lại & tại Ms Excel Ta nhấn Atl +{F8}  
- Cho hiện CS Project – VBAProject nếu nó chưa hiện (menu View->Project Explorer) -> Modules -> module1  
Tại CS này ta nhập một Mc đơn giản như sau:

Code:

**Sub MyMacro()**

Dim StrC as String

StrC = CHR(10) & CHR(13) & “XIN CHAO MUNG!”

MsgBox "My Macro!" & StrC, , ”9’ Da Qua!”

**End Sub**

Dòng lệnh có chứa CHR(13) chẳng qua là lệnh nối chuỗi ký tự. Bạn thử Mc làm việc cũng bằng cách đóng WorkBook lại & mở ra một cách bình thường như trên.  
Tại sao nói mở ra một cách bình thường?, vì nếu tại CS excel Ta ấn giữ phím SHIFT & mở file chứa Mc này thì Mc Workbook\_Open không được excel đếm xĩa đến!  
Ta nghiên cứu tiếp một Mc của WorkBook nũa, trước khi sang phần khác. Trước tiên ta phải chuột vô biểu tượng trái hơn của menu File & chọn dòng View Code;  
Nếu trong CS Object bên trái có dòng Workbook; thì ta tìm dòng NewSheet & ấn chọn nó. Cũng như trước đây. Một Mc sẵn có xuất hiện, chờ ta nhập lệnh vô:

**Code:**

**Private Sub Workbook\_NewSheet(ByVal Sh As Object)**

**End Sub**

Ở đây Sh là một biến đối tượng do VB khai báo và truyền cho Mc này. Tất nhiên nếu chúng ta cần dùng, chúng ta có thể khai báo & sử dụng loại biến này như thường, nhưng thật cân nhắc vì hao tiền tốn của lắm!  
Một biến đối tượng có thể chứa hình ành, chứa mảng giá trị, thâm chí chứa cả workbook của excel luôn!. . . Nhưng lúc cần ta sẽ xem xét sau; Còn giờ đây chúng ta nhập các dòng này vô giữa hai dòng lệnh trên

Code:

Application.displayAlerts = False

Msgbox “Sorry, you can’t add any more sheet to this workbook!”

Sh.delete

apPlication.displayalerts= -1

Dòng lệnh 1 phục vụ cho dòng lệnh 3: không cho hiện cảnh báo khi xóa Sh;  
Dòng lệnh cuối: phục hồi lại chế độ cảnh báo; Dòng còn lại bạn tự dịch lấy!  
Một vấn đề nữa theo tôi là mới xuất hiện, đó là ByVal. Đây, cũng như ByRef là những từ khóa chỉ ra biến được truyền là loại biến gì & đối xử với nó ra sao!  
Mặc định (nếu ta không có từ khóa như bài trước đây đã gặp) là truyền tham biến. Khi đó chương trình con nhận loại biến này có thể đổi hay gán cho nó giá trị khác. Ngược lại khai báo ByVal trước biến có nghĩa là chương trình cha gọi nó (chương trình con) truyền biến loại tham trị. Tuy chương trình con có thể dùng & biến đổi nó; nhưng khi kết thúc phần chương trình con thì biến này ở chương trình chính vẫn mang giá trị trước lúc truyền.

**3. Từ macro đến function**

Nếu từ CS excel ta ấn ATL+{F11} ta sẽ đến CS VBE (Visual Basic Editor). Nếu đây là một workbook mới, thì có thể chỉ xuất hiện CS Project – VBAProject phía bên trái màn hình; còn bên phải, phần chủ yếu của CS VBE có thể chưa có gì (ngoài một màu sám). Để có thể nhập trực tiếp một Mc , trước tiên ta vô menu Insert -> Module. CS soạn thảo của VBE thực sự hiện ra, nó cũng tương tự như SC mà ta đã thấy như đầu bài đã nêu.  
Ta nhập Mc sau:

Code:

**Sub FuncTion\_()**

Sheets("C0").Select

Range("E4").Value = Range("A4").Value + 9

**End Sub**

Sau đó ta lưu & ấn tổ hợp ALT+Q để đóng CS này lại; Trở về excel, ta lại vô menu Tool -> Macro->Macro để gán phím nóng cho Mc này bằng tổ hợp CTRL+SHIFT+F. Xong ta trở về trang tính, tại A4 ta nhập công thức: =INT(8 \* RAND()) & ENTER để ô A4 hiện một giá trị. Sau đó ta nhấn vài lần tổ hợp kích hoạt Mc. Sau mỗi lần nhấn tổ hợp phím ta xem thay đổi kết quả tại các ô mà Mc đề cập đến.  
Tiếp theo ta sẽ xây dựng hai Mc để giao & nhận giá trị giao , như sau:

Code:

Option Explicit

**Sub GiaoBien()** ‘

Dim So

So = InputBox("HAY NHAP MOT SO:")

If IsNumeric(So) Then FuncTion\_ (So)

**End Sub**

**Sub FuncTion\_(lSo As Long)**

Sheets("C0").Select:

Range("E4").Value = Range("A6").Value + lSo

**End Sub**

( Chỉ cần sửa Mc Function\_ lại để nó có thể nhận 1 biến số loại Long.)  
Sau khi ta nhập một giá trị bất kỳ vô ô A6 , ta để nháy vô trong vùng Mc GiaoBien & ấn {F5} để chạy 2 Mc & xem kết quả tại ô E4 trên sheets(“C0”)  
Chúng ta làm tất cả từ nãy đến giờ để làm gì?. Để thấy rằng chúng ta cần phải viết một hàm tự tạo là cần thiết. Vì Mc có thể nhận biến ta giao, nhưng nó chỉ cho ra kết quả tại ô cố định (đó là E4). Nếu ta viết hàm người dùng thì sẽ hiển thị kết quả nơi nào ta muốn; Thật vậy, Các bạn hãy nhập vô hàm sau:

Code:

**Function Function0( Bien0, Optional Bien7)**

If Bien7 = 0 then Bien7 = 9

Function0 = Bien0 + Bien7

**End Function**

Sau đó trở về trang tính; Tại một ô chưa có dữ liệu nào đó ta nhập dấu ‘=’ & vô menu Insert -> Function. Trong CS Or select a category ta chọn hàng User Defined. Trong CS Select a function ta sẽ thấy & chọn tên Function0. Đến đây bạn thao tác như những hàm của Excel khác; Sau khi chọn 2 ô dữ liệu làm đối số của hàm ta nhấn ENTER sẽ ra kết quả tại ô đang kích hoạt.  
Từ khóa Optional cho phép ta chọn ô thứ hai là ô không chứa dữ liệu, nhưng hàm Function0 biết cách tự xoay sở. Bạn hãy kiểm nghiệm phần này!

**Bài 10 : cách dùng VBA tương tự như trong các trường hợp ta vô menu Edit -> Goto -> Special**

**Phần giới thiệu nguồn tài nguyên trên mạng:**

**The SpecialCells Method in Excel VBA**

One of the most beneficial Methods in Excel (in my experience) is the SpecialCells Method. When used, it returns a **Range Object** that represents only those type of cells we specify. For example, one can use the SpecialCells Method to return a **Range Object** that only contains formulae. In fact, we can, if we wish, even narrow it down further to have our Range Object (containing only formulae) to return only formulae with errors.  
**The syntax for the SpecialCells Method is**;  
*expression*.SpecialCells(Type, Value)  
Where "*expression*" must be a Range Object. For example ***Range("A1:C100"), ActiveSheet.UsedRange*** etc.  
**Type=XlCellType and can be one of these XlCellType constants.**  
xlCellTypeAllFormatConditions. Cells of any format  
xlCellTypeAllValidation. Cells having validation criteria  
xlCellTypeBlanks. Empty cells  
xlCellTypeComments. Cells containing notes  
xlCellTypeConstants. Cells containing constants  
xlCellTypeFormulas. Cells containing formulas  
xlCellTypeLastCell. The last cell in the used range. Note this XlCellType will include empty cells that have had any of cells default format changed.  
xlCellTypeSameFormatConditions. Cells having the same format  
xlCellTypeSameValidation. Cells having the same validation criteria  
xlCellTypeVisible. All visible cells

These arguments **cannot** be added together to return more than one XlCellType.

**Value=XlSpecialCellsValue and can be one of these XlSpecialCellsValue constants.**  
xlErrors  
xlLogical  
xlNumbers   
xlTextValues  
These arguments **can** be added together to return more than one XlSpecialCellsValue.  
The SpecialCells Method can be used in a wide variety of situations when you only need to work with cells housing specific types of data. For example, the code below would return a Range Object representing all formulae on the active Worksheet.  
ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas)  
If we wanted, we could narrow this down further to only return a Range Object representative of all formulae that are returning numbers.  
ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas,xlNumbers)  
Once we have the specific Range Object type returned we can then work with only those cells. This can often be done in one line of code, or you may need to [**loop**](http://www.ozgrid.com/VBA/VBALoops.htm) through the range. See examples below;  
**Sub ColorAllFormulae()** ActiveSheet.UsedRange.SpecialCells \_  
 (xlCellTypeFormulas).Interior.ColorIndex = 36  
**End Sub  
Sub NegativeAllNumberFormula()**Dim rRange As Range, rCell As Range  
 Set rRange = ActiveSheet.UsedRange.SpecialCells \_  
 (xlCellTypeFormulas, xlNumbers)  
 For Each rCell In rRange  
 rCell = rCell.Value \* -1  
 Next rCell  
**End Sub**   
Although I have used a loop on the second macro, so that all returned numbers are converted to their negative counterparts, we could make use of PasteSpecial to do so without looping and allow the formulae to remain in the cells. That is;   
**Sub NegativeAllNumberFormula2()**

With Range("IV65536")

.Value = -1

.Copy

ActiveSheet.UsedRange.SpecialCells \_

(xlCellTypeFormulas, xlNumbers).PasteSpecial \_

xlPasteValues, xlPasteSpecialOperationMultiply

.Clear

End With

**End Sub**

**SpecialCells Gotcha!**

If you are familiar with Excel and it's built in features, such as SpecialCells, you will know that when/if one specifies only a single cell (via Selection or Range) Excel will assume you wish to work with the entire Worksheet of cells. For example, the 2 macros below would both select ALL blank cells on a Worksheet.

**Sub SelectAllBlanks()**

ActiveSheet.UsedRange.SpecialCells(xlCellTypeBlanks).Select

**End Sub**

**Sub SelectAllBlanks2()**

Range("A1").SpecialCells(xlCellTypeBlanks).Select

**End Sub**

So, as you can see, specifying only a single cell Range can give unwanted results.  
**SpecialCells for Formulae & Constants**While we cannot specify more than one XlCellType (e.g. xlCellTypeConstants+xlCellTypeFormulas would fail) we can use the SpecialCells method to return only used cells housing numbers on a Worksheet (formulae & constants) and omit any cells containing text (formulae & constants).

**Sub AllNummericCells()**

Dim rCcells As Range, rFcells As Range

Dim rAcells As Range

'Set variable to all used cells

Set rAcells = ActiveSheet.UsedRange

On Error Resume Next 'In case of no numeric formula or constants.

'Set variable to all numeric constants

Set rCcells = rAcells.SpecialCells(xlCellTypeConstants, xlNumbers)

'Set variable to all numeric formulas

Set rFcells = rAcells.SpecialCells(xlCellTypeFormulas, xlNumbers)

'Determine which type of numeric data (formulas, constants or none)

If rCcells Is Nothing And rFcells Is Nothing Then

MsgBox "You Worksheet contains no numbers"

End

ElseIf rCcells Is Nothing Then

Set rAcells = rFcells 'formulas

ElseIf rFcells Is Nothing Then

Set rAcells = rCcells 'constants

Else

Set rAcells = Application.Union(rFcells, rCcells) 'Both

End If

On Error GoTo 0

rAcells.Select

**End Sub**

You should take note of the **On Error Resume Next** statement in the above code. This is needed as when the SpecialCells Method condition cannot be met an error occurs. As you may, or may not know, a non valid Range Object returns the Nothing keyword. After Setting a Range [**variable**](http://www.ozgrid.com/VBA/variables.htm) to the SpecialCells Method we need to then check that we have been able to pass a Range Object to our Range variable. It is the If Statement (and 2 ElseIf) that checks this in the code above.

http://www.ozgrid.com/forum/showthread.php?t=61732&highlight=specialcells

**Phần giúp đỡ trong cửa sổ VBA:**Bạn vô CS (cửa sổ) VBA bằng tổ hợp fím ALT+{F11} & nhập các kí tự sau: [color =”Blue”] ‘ SpecialCells. Sau đó bôi toàn bộ tên phương thức & nhấn {F1} để được excel giải thích; Tuy nhiên giải thích này có thể làm chúng ta cảm thấy không đủ so với những gì đã ghi trên diễn đàn đã nêu như trên;

**Những áp dụng cụ thể   
A. Ta xem lại lần lượt các ví dụ nêu trên:**1.- Tô màu nền cho các ô trống trong vùng sử dụngGiả dụ trên trang tính trắng ta nhập‘Nam’ vô ô B2; ‘Thanh’ vô C3 & ‘Chanh’ vô ô D4& cho chạy macro **ColorAllFormulae** để biết xem những ô nào được tô màu nền?!  
2.- Đổi dấu dữ liệu của các ô công thức  
 Bằng cách sử dụng một trong hai macro đã nêu **NegativeAllNumberFormula** & **NegativeAllNumberFormula2**3.- Các trường hợp đếm các ô không chứa dữ liệuĐể hiểu rõ hơn về câu lệnhUsedRange.SpecialCells(x) ta xét đến ví dụ sau: Ta có trang tính trắng; Tại D4 ta nhập tên ‘Lê’; ô D5 nhập ‘Hoa’ & ô D6 ta nhập =D4 & “ “ & D5. Sau đó cho chạy macro sau:  **Sub CountAllBlanks()** ActiveSheet.UsedRange.SpecialCells(xlCellTypeBlanks).Select: MsgBox Selection.Count  
 Range("A1").SpecialCells(xlCellTypeBlanks).Select: MsgBox Selection.Count  
**End Sub**4.- Chọn vùng dữ liệu chứa công thức hay trị kiểu sốBằng macro **AllNummericCells**   
**B. Ta xem các phương án ứng dụng khác nhau trong 1 macro sau đây:  
Sub FillBlanks()**  
 Dim Rng As Range: Dim iZ As Integer  
 iZ = InputBox("CHON PHUONG AN:", , "9")  
 Select Case iZ  
 Case 1 ’a: Nhập trị ‘0’ cho toàn thể các ô trống trong vùng chọn   
 Range("C1:D9").SpecialCells(xlCellTypeBlanks).Value = 0  
 Case 2 ’b: Xóa giá trị các ô trên cột ‘A’ tương ứng với các ô trên cột B là trống   
 Columns("B:B").SpecialCells(xlCellTypeBlanks).Offset(0, -1).ClearContents  
 Case 3 ‘c: Ẩn/Xóa các dòng trong vùng từ 10 đến 40 khi ô trên cột A tương ứng là trống   
 Set Rng = Range("A10:A40").SpecialCells(xlBlanks).EntireRow  
 Rng.Hidden = True  
 ' Range("A10:A40").SpecialCells(xlBlanks).EntireRow.Delete  
 Case 4 ‘d: Hiện số dòng & địa chỉ của chúng, khi chúng không ẩn  
 Set Rng = ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Rows  
 MsgBox ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Rows.Count, , Rng.Address  
Case 5 ‘đ: Xóa giá trị các ô chứa số liệu, nhưng không là công thức   
 ActiveSheet.Cells.SpecialCells(xlCellTypeConstants, 23).ClearContents  
 Case 6 ‘e: Xóa giá trị các ô chứa số liệu trong vùng đặt tên ‘ConstantRef’   
 On Error Resume Next Range("ConstantRef").SpecialCells(xlCellTypeConstants).ClearContents  
 On Error GoTo 0  
case else  
 End Select  
**End Sub**

**C. Tự nghiên cứu macro sau đây:  
Sub CountBlanks()** Dim lCount As Long  
 On Error Resume Next  
 lCount = Range("C9:H9,L9:Q9").SpecialCells(xlCellTypeBlanks).Cells.Count  
 On Error GoTo 0  
 Select Case lCount  
 Case 0 To 1  
 Exit Sub  
 Case 2  
 'Do stuff  
 Case 6  
 'Do stuff  
 Case Else  
 'Do stuff  
 End Select  
**End Sub  
D. Các macro có thể gặp trong thực tế sau đây:  
1./ Ẩn hay xóa dòng dữ liệu theo các điều kiện** Một khi ta muốn xóa (hay làm ẩn đi) một số dòng dữ liệu theo nhiều điều kiện khác nhau ta cò thể nhờ đến Mcr (macro) **DeleteRowsWithSpecifiedData** sau đây.   
**Sub DeleteRowsWithSpecifiedData()**  
29 Dim Rng As Range  
1 Sheets("S3").Select: Columns(1).EntireColumn.Insert  
 2 With Range("A1:A" & ActiveSheet.UsedRange.Rows.Count)  
 3 .FormulaR1C1 = \_  
 "=IF(AND( OR(ISTEXT(RC[1]), ISBLANK(RC[1]), LEFT(RC[4],4)=""PAGE"") , LEFT(RC[5],5)<>""TOTAL""),NA(),FALSE)"  
4 .Value = .Value  
21 If Rng Is Nothing Then  
22 Set Rng = .SpecialCells(xlCellTypeConstants, xlErrors).EntireRow  
23 Else  
24 Rng = Union(Rng, .SpecialCells(xlCellTypeConstants, xlErrors).EntireRow)  
25 End If  
9 On Error Resume Next  
'11 .SpecialCells(xlCellTypeConstants, xlErrors).EntireRow.Delete  
 5 End With  
6 On Error GoTo 0  
28 Rng.Hidden = True  
7 Columns(1).EntireColumn.Delete  
**End Sub**Thủ tục nàygộpchung của hai phương án, đó là làm ẩn hay xóa dòng dữ liệu. Hiện tại thủ tục áp dụng trong phương thức ẩn các dòng dữ liệu thỏa các điều kiện. Khi muốn Mcr thực thi phương án xóa dòng dữ liệu ta phải:  
a./ Vô hiệu hóa các dòng lệnh có số lớn hơn 20 (bằng cách thêm dấu nháy đơn vô trước dòng lệnh)  
b./ Hiệu lực hóa dòng lệnh số 11 (bỏ đi dấu nháy đơn đầu dòng lệnh)  
Ta xét cụ thể trường hợp Mcr làm ẩn các dòng dữ liệu theo điều kiện nêu tại dòng lệnh 3, như sau:  
 Trước tiên, Mcr thêm 1 cột vô trang tính, trở thành cột A mới;  
 Tiếp theo là dòng lệnh yêu cầu khảo sát vùng A1:A(i), ở đây i là dòng cuối cùng của trang tính có chứa dữ liệu;  
(Dòng lệnh thứ 3 khỏi diễn dịch! Hơn nữa trong thực tế nó sẽ bị thay đổi theo yêu cầu cụ thể của ta)  
 Các dòng lệnh từ 21..25 nói rằng:  
 Nếu biến Rng chưa được gán trị thì gán bằng dòng có ô A(j) hiện hành đang chứa giá trị lỗi (NA())  
 Ngược lại, nếu Rng đã được gán thì thêm vô nó một dòng nữa (bằng phương thức UNION).  
 Dòng lệnh 28 làm ẩn vùng địa chỉ chứa trong biến Rng;  
Cuối cùng là xóa cột phụ mà Mcr đã tạo, để trả về trạng thái địa chỉ ô của trang tính ban đầu.

**2./ Khử lần lượt các ô trống trong 4 cột đầu tiên của dữ liệu   
Sub DeleteBlanks()** Dim intCol As Integer  
 For intCol = 1 To 4 [Color=”Blue” '\* \* \* Cols A to D   
 Range(Cells(2, intCol), Cells(333, intCol)).SpecialCells(xlCellTypeBlanks).Delete Shift:=xlUp  
 Next intCol  
**End Sub**Bắt đầu từ cột A, Mcr sẽ chép dữ liệu phía dưới lên ô trống một cách liên tục.  
Ví dụ cột A từ A2 đến A336 có hai ô trống thì sau khi Mcr chạy gần nữa đoạn đường, thì tại cột A dữ liệu sẽ có từ A2 đến A334  
3**./ Xóa các dòng thõa một trong hai điều kiện tại hai cột**Giả dụ ta có 1 cơ sở dữ liệu về học sinh. Cuối năm cần lập DS (danh sách) những em được lên lớp; DS chỉ là những học sinh tại thành phố (trường [TINH] tại cột ‘N’ hiện có các mã HCM, DN, LA, BD & BT); Học sinh ở lại lớp đã được ghi chú vô cột ‘O’. gồm có ‘HK Kem’, ‘O Lai’, ‘Nghi Hoc’ & ‘Bo Hoc’. Mcr dưới đây sẽ thực thi nhiệm vụ một cách hoàn hảo:  
**Sub XoaDongTheoDuLieu()**  
 Dim RngTest As Range, RngSelect As Range, rng As Range  
 'Fill any blank in column "O" with formula   
 Set RngTest = Intersect(ActiveSheet.UsedRange, Columns("N:O"))  
 RngTest.Columns(2).SpecialCells(xlCellTypeBlanks).FormulaR1C1 = "=rc[-1]"  
 'Selects all cells in columns "N" and "O" that match   
 For Each rng In RngTest  
 If rng = "LA" Or rng = "DN" Or rng = "BT" Or rng = "BD" Or rng = "HK Kem" \_  
 Or rng = "O Lai" Or rng = "Nghi Hoc" Or rng = "Bo Hoc" Then  
 If RngSelect Is Nothing Then Set RngSelect = rng  
 Set RngSelect = Union(RngSelect, rng)  
 End If  
 Next rng  
 'Deletes rows that match  
RngSelect.EntireRow.Delete  
**End Sub**   
Trong thủ tục có dùng hai phương thức Intersect & Union. Hai phương thức này mình đã có giới thiệu một bài trên diễn đàn. (Tìm kiếm bằng từ khóa Intersect)  
  
4**./ Biến các ô công thức trong vùng chọn thành dữ liệu   
Sub ValuesOnly()** Dim rRange As Range, RngSC As Range: Dim lCalc As Long  
 On Error Resume Next  
3 Set rRange = Application.InputBox(Prompt:="Select the formulas", Title:="VALUES ONLY", Type:=8)  
4 Set RngSC = rRange.SpecialCells(xlCellTypeFormulas)  
 If RngSC Is Nothing Then Exit Sub  
6 Set rRange = Application.Intersect(rRange, RngSC)  
 If rRange Is Nothing Then Exit Sub  
 On Error GoTo 0  
9 With Application  
 .ScreenUpdating = False: lCalc = .Calculation  
 .Calculation = xlCalculationManual  
12 End With  
 Set RngSC = Nothing  
14 For Each RngSC In rRange  
 RngSC = RngSC.Value ' rRange = rRange.Value  
16 Next RngSC  
17 With Application  
 If Not lCalc = 0 Then .Calculation = lCalc  
 .ScreenUpdating = True  
20 End With  
21 Set rRange = Nothing: Set RngSC = Nothing  
**End Sub**   
Câu lệnh 3 yêu cầu ta nhập vô chuỗi biểu thị vùng dữ liệu cần thiết biến từ dạng công thức sang dạng dữ liệu, ví dụ: ‘A1:B99’ hay ‘A$9:$B500’ hoặc ‘$A$2:$U$78’  
Câu lệnh 4: Các ô chứa công thức trong vùng chọn trên được đem gán vô biến RngSC đã khai báo.  
Câu lệnh 6: Các ô nào thõa với phương thức Intersect giữa ô công thức (c 4) & vùng chọn ban đầu (c3) được gán lại vô biến rRange;  
Các câu lệnh 5, 7 dùng để thoát khỏi chương trình khi không có ô nào thõa điều kiện;  
Câu lệnh 15 chính là câu biến các ô chứa công thức thành các ô chứa trị;  
Cuối cùng, câu lệnh 21 dùng để báo là chúng ta không xài các biến thuộc loại Range nữa; Mục đích chủ yếu là thu hồi bộ nhớ, tránh lãng phí tài nguyên.  
5**./ Sao chép vùng lọc sang một Sheet mới**Trong Mcr sau chúng ta có dịp xem xét đến cách sử dụng phương thức SpecialCells đối với trang tính đã qua lọc. Sau khi thực hiện lọc từ các dòng dữ liệu, chúng ta muốn chuyển chúng sang một sheet khác để làm báo cáo, thì Mcr CopyAFilter sẽ phục vụ bạn một cách miễn phí:   
**Sub CopyAFilter**()  
 Dim Rng As Range   
 With Sheet3  
 If Not .FilterMode Then  
 MsgBox "AutoFilter?": Exit Sub  
 End If  
 Set Rng = .AutoFilter.Range.Offset(1, 0).Resize(.AutoFilter.Range.Rows. \_  
 Count - 1).SpecialCells(xlCellTypeVisible)  
 'set a range = to visible cells (excluding the header)  
 Rng.Copy Destination:=Sheet4.Range("A1")  
 End With  
**End Sub**  
  
6**./ Các bạn tự tìm hiểu 3 Mcr này:  
  
Sub DeleteStuff()**  
 Dim Rng As Range  
 Set Rng = Range("Q2:Q" & Cells(65536, "Q").End(xlUp).Row)  
 If ActiveSheet.AutoFilterMode Then Cells.AutoFilter  
 Columns("Q").AutoFilter Field:=1, Criteria1:="="  
 On Error Resume Next 'in case nothing found to delete  
 Rng.SpecialCells(xlCellTypeVisible).EntireRow.Delete  
 On Error GoTo 0  
 ActiveSheet.AutoFilterMode = False  
End Sub   
Sub MAreas1()   
 If ActiveSheet.FilterMode = True Then 'Select first row below heading  
 ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Areas(2).Select  
 ' \* Will tell you the actual row number of the second visible row  
 MsgBox ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Areas(2).Rows(1).Row  
 End If  
End Sub  
Sub MAreas2()  
 If ActiveSheet.FilterMode = True Then 'Select cell in 3rd row first column  
 ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Areas(3).Columns(1). \_  
Cells(1, 1).Select  
 ' \* Will tell you the actual row number of the third visible row  
 MsgBox ActiveSheet.UsedRange.SpecialCells(xlCellTypeVisible).Areas(3).Rows(1).Row  
 End If  
End Sub

**Bài 11. Tìm hiểu các phần tử tập hợp**

Nếu các bạn chưa có khái niệm về toán cao cấp thì khi đọc những trang này cần liên hệ mật thiết với các vật thể tồn tại quanh ta. Từ đó sẽ giúp chúng ta cùng nhau hiểu các phần tử trong excel dễ hơn.

[B]1./ Phần tử là gì? [/b]

*[I]a./ Phần tử thuộc về tập hợp*:[/I] Trong thế giới quanh ta, thì thành phố, cơ quan, công sở, trường học, phân xưỡng, công nhân viên. . . đều là các phần tử. Phần tử có thể chứa các loại phần tử khác. Tỷ như thành phố có nhiều trường & trong trường có nhều lớp học, trong lớp lại có nhiều học sinh.

Mỗi nhân viên là một phần tử thuộc về một loại phần tử khác: công sở (cơ quan). Để xác định một nhân viên, bạn có thể chỉ ra tên ( anh Trần An), chỉ vị trí (đồng nghiệp ngồi góc phải cuối phòng), chỉ tính chất (bạn mặc áo xanh & đội nón màu cam). Tuy nhiên tập hợp tất cả các nhân viên cũng là một phần tử.

Ta mở chương trình excel 2003, để xuất hiện một bảng tính trắng duy nhất. Nói là duy nhất, vì nếu bạn còn bảng tính nào khác thì nên đóng nó lại (dùng lệnh Unhide trong menu window để áp dụng cho cả bảng tính ẩn)

Trong excel cũng vậy, mỗi bảng tính là một phần tử. Bạn có thể chỉ ra bảng tính bằng tên (mở BaiTap.XLS), bằng vị trí (kích hoạt bảng tính thứ ba trong DS (danh sách)), chỉ trực tiếp (khi lưu bảng tính hiện hành). . . .tập hợp các bảng tính cũng là một phần tử thuộc loại khác, ta tạm gọi là phần tử ‘các bảng tính’.

(ũng đúng như vậy đối với các trang tính, các vùng, các ô trong trang tính. . . .

*[I]b./ Phần tử có tính chất:[/I]* Hãy nhớ lại hình ảnh của bạn trước đây một con giáp: một cô/cậu bé cao khoảng 1,55m, bận áo trắng, ngồi giữa dẫy trái của lớp học, tóc không dài & màu đen, mắt không mở. . .

Chiều cao, màu áo, vị trí, màu & độ dài tóc, trạng thái mắt. . . là các tính chất của phần tử. Cô bạn kế bên cũng có các tính chất đó, vì cô ấy cũng là phần tử thuộc loại học sinh. Chiếc ghế cô ta ngồi cũng có tính chất chiều cao, vị trí, cân nặng, nhưng không có tính chất về độ dài tóc, nhưng ngược lại, nó có tính chất nơi sản xuất mà bạn & cô kế bên không có. Như vậy cái ghế là phần tử thuộc loại khác, và các phần tử thuộc các loại khác nhau không có tính chất giống hệt nhau.

Tập hợp ‘các học sinh trong lớp’ học cũng là một phần tử. Nó không có tính chất về chiều cao, cân nặng. . . , nhưng nó có tính chất khác: Tổng số, số nam, số nữ, số có mặt. . . . Như vậy, phần tử ‘các học sinh’ thuộc loại khác với phần tử học sinh vì chúng có tính chất khác nhau.

Một số tính chất có thể thay đổi được. Bạn thay đổi trạng thái tình cảm khi sếp khen/chê trước tập thể, màu tóc của bạn sau một con giáp nữa sẻ không còn đen. . . Nhưng cũng có những tính chất là bất di bất dịch: Ngày sinh (trong phạm vi hoạt động bình thường của tập hợp ‘lớp’ hay ‘trường’ điều này là không khả thi).

(ác phần tử trong excel cũng có tính chất. Mỗi bảng tính đều có ngày sinh, ngày hiệu chỉnh lần cuối, do một vĩ nhân, hay do một tên ngốc nào đó tình cờ lập ra . mỗi hàng, mỗi cột, đều có chiều cao & độ rộng. Mỗi menu đều có tiêu đề, người ta liệt kê số trang cho mỗi trang tính; một số tính chất rất dễ bị tổn thương như tên của trang tính, nhưng một số khác lại khó hơn: thay đổi tổng số ô trong một trang tính.

*[I]c./ Phần tử có phương pháp:[/I]* Mỗi nhân viên đều có thể đọc sách & đọc tài liệu, mỗi sinh viên đều có thể gõ & nhập hàm =DSUM() vô ô trang tính. Ở một thời điểm nào đó có thể nhân viên này đang đọc thì dồng sự khác đang viết thư điện tử. . .Nhưng quan trọng là vấn đề có khả năng đọc & có khả năng gởi thư điện tử. Những hành động đọc, viết nầy nọ đó ta gọi là phương pháp của phần tử đó (có nhiều người hay dùng phương thức thay vì phương pháp)

Phương pháp làm thay đổi tính chất. Khi Lê Chanh thực hiện phương pháp đổ nước lên đầu mình, thì tính chất chiếc áo sẽ bị ướt.

Một tập hợp các phần tử có các phương pháp khác với các phương pháp của từng thành phần của nó. Khi một nhân viên mới vô thử việc, thì phương pháp thêm vào là phương pháp của phần tử ‘cơ quan’ không thể bất cứ phần tử ‘nhân viên’ riêng rẽ nào thực hiện được phương pháp này.

Một phương pháp quan trọng của phần tử tập hợp là chỉ ra một phần tử thuộc về nó. Khi giảng viên gọi: “Em Thanh, lên bảng!” Tác động gọi ra một phần tử trong một tập hợp & do đó tạo ra một liên hệ với phần tử đó là một phương pháp của các phần tử, chứ tuyệt nhiên không phải là phương pháp của phần tử ‘Em Thanh’ này.

Ở đây là phải hiểu là giảng viên gọi với cả lớp, chứ không phải chỉ tác động lên phần tử ‘Thanh’. Vì một hiển nhiên là cả lớp đều nghe cô gọi & thấy kết quả của phương pháp gọi này. (Hơn nữa nếu chỉ có giảng viên & ‘Thanh’ thì chưa chắc giảng viên gọi như vậy!)

Mỗi tập hợp các phần tử trong excel đều có phương pháp Add để thêm các phần tử cho mình. Trong trang tính có phương pháp Calculate để tính lại giá trị trong các ô. Đồ thị có phương pháp ChartWizard để thay đổi nhanh chóng các tính nết của đồ thị.

Sự phân biệt giữa phương pháp & tính chất cũng rất mong manh. Khi bị Sếp mắng, nhân viên ‘Hoa’, ‘Hoa’ thực hiện phương thức trừng mắt (một hành động), hay cô ta gán một giá trị mới cho tính chất của mắt (tính dãn đồng tử). Cũng có lúc ta sẽ gặp khi một phương pháp trả về một giá trị.

Hãy xem phương pháp Intersect sau:[code][B]

Function AntiRange(LRng As Range, NRng As Range) As Range [/B]

Dim NewRng As Range, CurrCell As Range

For Each CurrCell In LRng.Cells

If Intersect(CurrCell, NRng) Is Nothing Then

If NewRng Is Nothing Then

Set NewRng = CurrCell

Else

Set NewRng = Union(NewRng, CurrCell)

End If

End If

Next

Set AntiRange = NewRng

Set NewRng = Nothing : Set CurrCell = Nothing [B]

End Function[/B] [/code]

[B]2./ Tìm hiểu các bảng tính [/B]

*a./ Thêm một bảng tính:* Từ bảng tính đang mở, ta sẽ có ít nhất là hai cách để mở CS (cửa sổ) Microsoft VB (Visual Basic), như sau:

* Phải chuột vô vùng trống trên thanh Toolbar & chọn từ DS đỗ xuống mục Visual Basic, khi thanh công cụ này hiện ra ta chọn biểu tượng Visual Basic Editor
* Nhấn tổ hợp phím Atl + {F11}

Từ CS này ta vô menu View -> Immediate để có thêm một CS ‘nghiệp vụ’ mới. Tiến hành thu nhỏ CS MVB (Microsoft Visual Basic) còn phân nữa để thấy được CS workbook bên dưới.

Trong CS Immediate, ta gỏ nhập câu lệnh:

[code]WorkBooks.Add [/code]

Và ấn ENTER. Bạn nên tìm đâu đó trên màn hình kết quả việc VBA thực hiện lệnh của bạn. Phương pháp Add đã thêm một phần tử bảng tính mới vố tập hợp các bảng tính của bạn.

/(/hư vậy là vùng Immediate này hữu dụng để chúng ta tìm hiểu về thành phần, phương thức . . . của VBA nói riêng & Excel nữa.

/(/ếu thích bạn có thể lặp lại lệnh này, bằng cách giản tiện hơn, đó là ấn trỏ chuột vô dòng lệnh trên & ấn ENTER một lần nữa.

*b./ Đếm số bảng tính*: Chúng ta vừa áp dụng phương pháp Add lên phần tử ‘các bảng tính’. Phần tử này cũng có những tính chất. Một trong những tính đó là đếm.

Trong CS Immediate ta gỏ nhập lệnh:

[code]?workbooks.count [/code]

Và thực hiện ấn ENTER xuống dòng. Tức thời tại dòng hiện hành xuất hiện con số chỉ ra số bảng tính có trong tập hợp

[QUOTE]CS Immediate còn là nới thoải mái vọc của dân tập tễnh đến VBA, này nha:

Bạn vừa tạo ra một kiệt tác UDF vừa ý ư, thì hãy đem đến đây thử các kết quả hàm trả về;

Bạn muốn kiểm nghiệm các hàm trong VBA ư, nơi này cũng là nơi tốt, chẵng hạn:

?1 + Date() [/QUOTE] và ấn ENTER xem sao(?)

/(/hư vậy ta thấy, muốn đếm số bảng tính ta dùng phương thúc Count, nhưng phương thức này không làm tăng/giảm số bảng tính; muốn tăng ta dùng phương thức Add. Như vậy, tính chất Count thuộc loại tính chất chỉ đọc (read-only), không khác mấy với tính chất ngày sinh tháng đẻ của phần tử ‘nhân viên’ nào đó.

*c./ Đóng bảng tính*: /(/goài phương pháp Add, phần tử WorkBooks có thêm phương pháp Close để đóng bảng tính. Gỏ vô CS Immediate lệnh [Code] WorkBooks.Close [/code] và nhấn ENTER, trả lời ‘No’ khi được hỏi điều gì đó.

/(hi đó mọi bảng tính đều biến mất. Như vậy phương pháp Close tác động lên phần tử WorkBooks (tập hợp các bảng tính); Có nghĩa là nó đóng toàn bộ tập hợp này.

Bạn thử thực hiện lại lệnh ?WorkBooks.Count xem sao (?)

/(/hư vậy rõ một điều, tính chất Count đã bị phương pháp Add & Close thay đổi không thương tiếc.

*d./ Chỉ định một bảng tính*: Phương pháp Close như trên đôi khi bất lợi, nếu khi ta chỉ cần đóng một hay một vài bảng tính cụ thể nào đó. Giống như giảng viên muốn gọi em Chi lên bảng, thì phải chỉ mặt đặt tên đó là phần tử nào trong tập hợp đang có.

Đến đây bạn nên thực hiện không dưới 7 lần lệnh WorkBooks.Add để tạo nhiều bảng mới.

Là CS soạn thảo nên trong Immediate ta có thể nhấn cùng lúc CTRL+{ENTER} để làm gì bạn thấy ngay đấy; Gỏ nhập vô dòng trống câu lệnh sau: [Code] ?WorkBooks.Item(1).Name [/code] và ấn ENTER. Tại dòng đối thoại sẽ xuất hiện một cái tên mà bạn tự suy ngẫm.

Theo như quyễn ghi chép “Chập chững đến VBA” ta đọc câu lệnh trên từ phải sang trái như sau:

Tên của phần tử thứ nhất của ‘các phần tử’ bảng tính là gì(?)

Từ ‘Name’ là tính chất của một phần tử bảng tính (WorkBook)

Bạn xem thử phần tử WorkBooks có tính chất này không (?)

Tiếp theo bạn gỏ [code] Workbooks.Item(2).close [/code] và ấn ENTER.. Sẽ có một bảng tính nào đó bị biến khỏi tầm mắt.

/(/ếu giờ bạn nhập lệnh [code] ?WorkBooks.Item(2).Name [/code] để xem hai tên được gọi có liền kề hay cách rời nhau không?

Item là một phương pháp mà mọi phần tử tập hợp đều chấp nhận. Phương pháp này đặt liên hệ đến một phần tử được xác định rõ của tập hợp đó. Có thể chỉ ra vị trí trong tập hợp đó (như các ví dụ trên), có thể chỉ bằng tên cụ thể ( sẽ chờ một chút tiếp sau)

Trong các câu lệnh VB, bạn không thể dùng cách bỏ chữ ‘s’ để chỉ phần tử trong tập hợp (chẳng hạn bỏ chữ ‘s’ trong chữ WorkBooks để chì phần tử riêng rẽ nào của ‘các phần tử’ workbooks). Muốn chỉ một phần tử cụ thể nào ta phải dùng phương pháp Item; Như để đóng workbooks thứ 5 ta dùng lệnh WorkBooks.Item(5).Close (không thể dùng Workbook(5).Close)

*e./ Xác định một bảng tính bằng tên:*

Chúng ta vừa mới xác định một bảng tính bằng vị trí của nó trong một tập hợp (Là vị trí tuần tự phát sinh của nó). Một phần tử nhất thiết phải được xác định theo tên của nó trong một tập hợp. Cho rằng để tiếp tục bạn cần phải tạo vài ba bảng tính đang hiện hữu.

Dùng thanh cuốn dời đến cuối CS Immediate, lựa tên 1 bảng tính ở giữa dãy bảng tính bạn đang có (VD là Book9). Gỏ dòng lệnh [Code]WorkBooks.Item(“Book9”).Activate [/code] và ấn ENTER. Dưới tác động của lệnh này Book9 sẽ được điều động lên nằm đầu dãy các bảng tính.

Activate là một trong những phương pháp của một phần tử bảng tính. Và quan trọng ở đây là ta đã chỉ ra phần tử đó bằng tên của nó. Chúng ta thấy, khi dùng tên, phải để tên trong dấu ngoặt kép. Bạn hãy tự mình dùng câu lệnh cho biến phần tử này khỏi tầm mắt của chúng ta.

Sau đó thi hành lệnh Activate nêu trên, thì Excel sẽ dận dỗi & báo lỗi có còn bảng tính ‘Book9’ nữa đâu mà kích hoạt nó!

*f./ Làm việc với bảng tính đang được kích hoạt:*

Giả sử cần thực hiện việc gì đó lên bảng tính đang kích hoạt (hiện hành) nên ta có thể tác động lên nó bằng những lệnh hơi khác hơn, ví dụ [Code] ActiveWorkBook.Close [/Code] sẽ làm cho bảng tính này sẽ đi vô dĩ vãng.

*g./ Thay đổi giá trị tính chất của bảng tính:*

Các tính chất Count & Name của phần tử workBooks & WorkBook tương ứng là các tính chất chỉ đọc. Các tính chất này bạn có thể xem được nhưng sẽ không cho bạn trực tiếp thay đổi giá trị của chúng.

Nhưng cũng có những tính chất thuộc loại đọc-ghi cho phép bạn thay đổi giá trị của chúng. Tính chất Saved của phần tử workbook là một tính chất như vậy. Trong CS Immediate ta tạo thêm tối thiểu 3 workbooks bằng cách ấn 3 lần dòng lệnh WorkBooks.Add

Sau đó ta thêm các dòng lệnh sau: [Code] WorkBooks.Item(1).Activate

Range(“B2: C4”).Value=”N2”

ActiveWorkBook.Close [/Code]

Dòng lệnh thứ hai ta đã gán giá trị vô một số ô của bảng tính hiện hành;

Sau khi ấn ENTER của dòng lệnh thứ ba, thì Excel sẽ hỏi ta có lưu những thay đổi vừa qua hay không? Câu trả lời chúng ta là không! (Nhưng nhớ rằng là chúng ta đã được hỏi thăm!)

Nhưng để Excel không hỏi gì cho tốn hao thời gian & tiền của, thay vì câu lệnh 3 ngắn như trên ta thêm cho dài ra một chút. Và để đối chứng ta nhập & thực hiện lại các dòng lệnh như sau:

[Code] WorkBooks.Item(1).Activate

Range(“B1: B2”).Value=”L.T. Thơm, Kiến An N0 -17”

ActiveWorkBook.Close SaveChanges := False [/Code]

Sau khi ấn ENTER, chúng ta có bị hỏi han nữa không vậy?!

Một điều nên nhớ là nếu chúng ta đã cho macro làm gì đó trên bảng tính. Sau đó muốn đóng lại thì nên ấn định giá trị thuộc tính lưu cho excel hiểu việc mình phải làm, trách để Excel hỏi han lại lôi thôi. Vì trong nhiều trường hợp sau này chúng ta không biết mình đã làm gì để bị hỏi han!

**Bài 12:KHÁI NIỆM VỀ PHẦN TỬ RANGE**

[code]Phần này dựa chủ yếu vô bài viết của chị HANDUNG107 tại :

<http://www.giaiphapexcel.com/forum/showthread.php?t=52> [/code]

Là một ô, một hàng, một cột, hoặc có thể là một vùng chọn của các ô, chứa một hay nhiều vùng ô kề nhau, hay 3-D range

**Cách thức sử dụng phần tử range**

Các tính chất (thuộc tính) và phương thức (phương pháp) sẽ trả về phần tử range được liệt kê như sau đây:

* Thuộc tính Range (**Range** property )
* Thuộc tính Cells (**Cells** property )
* Range and Cells
* Thuộc tính Offset **(Offset** property )
* Phương thức Union (**Union** method )

Chúng ta sẽ xét đến từng thuộc tính & phương thức nêu trên cũng như một số thuộc tính có họ hàng với chúng.

## Thuộc tính Range

Chúng ta sử dụng **Range**(*arg*), ở đây *arg* là tên của range, sẽ được trả về một đối tượng **Range** là một ô đơn (cell) hay vùng các ô.

Ví dụ sau đây sẽ gán giá trị của ô B1 vô ô C5

[code] Worksheets("Sheet1").Range("C5").Value = \_

Worksheets("Sheet1").Range("B1").Value [/code]

Ví dụ tiếp theo sẽ gán các số ngẫu nhiên từ 1 đến 9 vô vùng ô từ D1:F8. [Code]

Worksheets("Sheet1").Activate

Range("A1:H8").Formula = "= 1 + 8 \* Rand()" ‘Range is on the active sheet [/code]

Còn ví dụ thứ ba sau, sẽ xóa dữ liệu trong vùng có tên là Criteria. [Code]

Worksheets(1).Range("Criteria").ClearContents [/Code]

Nếu ta sử dụng đối số dạng text trong địa chỉ vùng, chúng ta chỉ được dùng loại kí hiệu A1 (chúng ta không thể dùng loại R1C1)

## Thuộc tính Cells

Ta sử dụng thuộc tính **Cells**(*row*, *column*), ở đây *row* là chỉ số hàng & *column* là chỉ số cột, sẽ được trả về một ô đơn. Ví dụ ta gán giá trị 34 vô ô B2. [Code]

Worksheets(1).Cells(2, 2).Value = 34 [/code]

Hay chứa tổng 1 vùng vô ô A2, như sau:. [Code]

ActiveSheet.Cells(2, 1).Formula = "=Sum(B1:B5)" [/Code]

Ví dụ dưới đây sẽ lập bảng tiêu đề cột & tiêu đề hàng cho một trang tính. Qua ví dụ này, chúng ta cũng thấy được sự khác biệt nho nhỏ trong cách dùng thuộc tính Range & thuộc tính Cells

. [php] Option Explicit

Sub SetUpTable()

Dim TheYear As Byte, TheQuarter As Byte

Worksheets("Sheet1").Activate

For TheYear = 1 To 5

Cells(1, 1 + 2 \* TheYear).Value = 2000 + TheYear

Next TheYear

For TheQuarter = 1 To 4

Range("B" & TheQuarter + 2).Value = "Q" & TheQuarter

Next TheQuarter

End Sub [/php]

Nếu ta sử dụng dạng thức *expression*.**Cells**(*row*, *column*) , mà ở đâu biến *expression* là một biến được trả về là một đối tượng **Range**, và *row* & *column* có mối liên hệ với góc trái nhất của Range, thì sẽ trả về một phần của Range. Ví dụ sau đây sẽ áp đặt công thức vô ô D5. [code]

Worksheets(1).Range("C5:F9").Cells(1, 2).Formula = "=Rand()" [/code]

## Range and Cells

Nếu sử dụng **Range**(*cell1, cell2*), mà *cell1* & *cell2* là đối tượng **Range,** đặt trưng của các ô đầu & ô cuối, thì sẽ trả về là 1 đối tượng **Range**. Ví dụ sau đây sẽ tạo đường viền cho các ô B2:I9. [Code]

With Worksheets(1)

.Range(.Cells(2, 2), .Cells(9, 9)) \_

.Borders.LineStyle = xlThick

End With [/Code]

Tiếp theo, chúng ta khảo sát đoạn mã lệnh sau. Yêu cầu sau khi khảo sát, bạn hãy cho biết biến Rng chấp nhận chưỗi nào trong các chuỗi sau:

‘A1B5’; ‘A:A’; ‘1:1’; ‘B:X’; ‘Ax1:A1’; ‘5:9’; ‘$A4:b$7’; ‘4 4’

[php]Sub RangeFromInputbox()

Dim Rng As Range

Set Rng = Application.InputBox \_

(Prompt:="Select any range", Title:="Demo", Type:=8)

MsgBox Rng.Address

End Sub[/php]

Thêm một ví dụ nữa, để thấy việc dùng các thuộc tính đang nêu:[code]

Sub LastRowAndColumn()

Dim lRow As Long, iCol As Integer

lRow = Range("A65432").End(xlUp).Row

iCol = Cells(2, 255).End(xlToLeft).Column

MsgBox Cells(lRow, 1).Address, , Range(Chr(64 + iCol) & 2).Address

End Sub[/code]

(Chúng ta cho macro chạy trên một trang tính có dữ liệu để xác định tính đúng đắn của macro này!

Có thể không thích xái thuộc tính range ta xài thuộc tính Cells, và ngược lại? Ta xét đến ví dụ sau:

[php]Sub AllLoop()

1 Dim Clls As Range

2 For Each Clls In Cells

3 If Clls.Value = "@" Then

4 Clls.Activate: Exit For

5 End If

6 Next Clls

End Sub [/php]

Vậy bạn thử đổi chữ ‘Range’ trong dòng lệnh 1 thành ‘cells’ xem sao. Sau khi ta nhập xong, VBE cũng chuyễn chữ này thành ‘Cells’, Nhưng khi bấm chạy macro thì nhận được thông báo lỗi!

Tương tự, nếu ta thay chữ ‘Cells’ của dòng lệnh 2 bằng chữ ‘range’. Thì cũng nhận được kết quả tồi tệ không kém,

## Thuộc tính Offset

Sử dụng thuộc tính **Offset**(*row, colu*mn), mà khi đó *row* & *column* là số hàng & số cột khác lệch của range được thuộc tính trả về so với địa chỉ nguyên thủy trước đó. Để rõ hơn ta xét ví dụ sau: [php]

Worksheets("Sheet1").Activate

'Can't select unless the sheet is active’

Selection.Offset(3, 1).Range("A1").Select [/php]

Chúng ta xét thêm một ví dụ nữa, sau đây: [code]

Sub OffsetRanges()

MsgBox Range("A2:B3").Offset(2, 4).Address, , Range("A2:B3").Address

End Sub [/code]

Các bạn cho macro chạy & tự rút cho mình kết luận cần thiết.

Cũng giống trong excel, chúng ta có thể không ghi 1 trong 2 biến của thuộc tính này. Trong VBA chúng ta có thể thấy các dòng lệnh sau

[code] Set Rng = Range(“B” & Range(“B65432”).End(xlUp).Row).Offset(1)

Temp = Range(“C1”).Offset(,Range(“B65432”).End(xlUp).Row) [/code]

## Phương thức Union

Sử dụng phương thức **Union**(*range1, range2*, ...) để được trả về một hỗn hợp vùng — là vùng gộp chung giữa hai hay nhiều hơn những khối ô. Thí dụ sau đây sẽ tạo ra đối tượng xác định bỡi phương thức Union hai vùng A2:B3 and C3:D4, và sau đó chọn nó. [php]

Sub UnionMethod

Dim r1 As Range, r2 As Range, myMultiAreaRange As Range

Worksheets("sheet1").Activate

Set r1 = Range("A1:B2"): Set r2 = Range("C3:D4")

Set myMultiAreaRange = Union(r1, r2)

myMultiAreaRange.Select

End Sub[/php]

Nếu chúng ta cần làm việc với vùng chọn, mà vùng này bao gồm hơn 1 vùng riêng rẽ, tốt hơn ta dùng thuộc tính **Areas**. Về thuộc tính Areas xin chưa đi sâu trong bài viết này!

Để hiểu rõ hơn về phương thức Union, xin các bạn tham khảo thêm bài viết tại:

<http://www.giaiphapexcel.com/forum/showthread.php?t=480> ;

Có bạn trên diễn đàn chúng ta đặt câu hỏi:

Có 2 vùng: A=Range("A1:G20"); B=Range("C3:D4")  
Làm sao xác định được vùng C=A-B (là vùng A mà loại bỏ các ô trong vùng B) bằng VBA ?

Tôi xin giới thiệu một trong những cách đó là dùng phương thức UNION(), như sau: [php]

Option Explicit

Sub Range11()

Dim Rng As Range, RngA As Range, RngB As Range, RngC As Range

Set RngA = Range("A1:E16")

Set RngB = Range("B7:C10")

For Each Rng In RngA

If Intersect(Rng, RngB) Is Nothing Then

If RngC Is Nothing Then

Set RngC = Rng

Else

Set RngC = Union(RngC, Rng)

End If: End If

Next Rng

MsgBox RngC.Address

End Sub [/php]

(Đoạn mã trên dùng để xét từng ô trong toàn bộ các ô của vùng A; Khi gặp 1 ô nào đó không thuộc vùng B thì gán vô biến vùng C bằng phương thức union(). Như vậy để hiểu thật kỹ cách thức trong khi thực thi C = A – B ta phải hiểu thêm phương thức INTERSECT(). Mong các bạn tự tham khảo thêm)

## Thuộc tính Resize

Nếu trong cửa sồ (CS) VBE (bằng cách nhấn các phím Atl+{F11} hay từ CS Immediate ta nhập chữ Resize, bôi chọn toàn bộ và nhấn {F1}, phần trợ giúp của excel về đặt tính này như sau:

Resizes the specified range. Returns a **Range** object that represents the resized range.

*expression****.Resize(RowSize****,* ***ColumnSize)***

*expression    Required. An expression that returns a* ***Range*** *object to be resized.*

***RowSize****Optional* ***Variant****. The number of rows in the new range. If this argument is omitted, the number of rows in the range remains the same.*

***ColumnSize****Optional* ***Variant****. The number of columns in the new range. If this argument is omitted, the number of columns in the range remains the same.*

***Example***

*This example resizes the selection on Sheet1 to extend it by one row and one column.*

*Worksheets("Sheet1").Activate*

*numRows = Selection.Rows.Count*

*numColumns = Selection.Columns.Count*

*Selection.****Resize****(numRows + 1, numColumns + 1).Select*

*This example assumes that you have a table on Sheet1 that has a header row. The example selects the table, without selecting the header row. The active cell must be somewhere in the table before you run the example.*

*Set tbl = ActiveCell.CurrentRegion*

*tbl.Offset(1, 0).****Resize****(tbl.Rows.Count - 1, \_*

*tbl.Columns.Count).Select*

## (Xin phép cho tôi miễn dịch phần vừa rồi, vì có khi dịch không tốt bằng nguyên bảng). Để hiểu rõ hơn về thuộc tính này, ta xét thêm vài ví dụ sau đây:

## Ta có macro (Mc) dùng để mở rọng vùng chọn, như sau [php]

## Sub ExpandingArray()

## Dim xRng As Range, TrRng As Range

## Set xRng = Range("c2:d9"): Set TrRng = Range("b4:c35")

## Set xRng = xRng.Resize(xRng.Rows.Count, xRng.Columns.Count + 5)

## MsgBox xRng.Address, , "A"

## Set TrRng = TrRng.Resize(xRng.Rows.Count + TrRng.Rows. \_

## Count, TrRng.Columns.Count)

## MsgBox TrRng.Address, , "B"

## End Sub [/php]

## Các bạn cho Mc chạy, khi đó lần lượt xuất hiện 2 hộp thoại, cho ta thông tin địa chỉ vùng đã được mở rộng dưới tác động của đặc tính Resize.

## Đến đây tôi thấy cần phân biệt giữa hai thuộc tính OFFSET() & RESIZE()

## Chúng ta xem xét đến macro sau: [php]

## Sub OffsetAndResize()

## Dim Rng As Range

## Set Rng = Range(“B2”).Offset(, 1)

## MsgBox Rng.Resize(2,2).Address,, Rng.Address

## End Sub [/php]

## Sau khi ta đọc dữ liệu trong hộp thoại & đối chiếu với các chỉ số trong các thuộc tính, ta sẽ rạch ròi hơn xíu về cách thức của 2 thuộc tính trên.

## Thiết nghĩ cũng cần nêu lại vấn đề danh sách tiêu đề của vùng dữ liệu như ví dụ sau

## Sub TableDataHeaders()

## Dim rTable As Range

## Set rTable = Sheet1.Range("A1").CurrentRegion

## Set rTable = rTable.Resize(rTable.Rows.Count - 1)

## MsgBox rTable.Address, , "1"

## Set rTable = rTable.Offset(rTable.ListHeaderRows)

## MsgBox rTable.Address, , "2"

## End Sub

## Macro này ta đã gặp khi nghiên cứu về thuộc tính CurrentRegion. (bạn nào cần xen lại, có thể nhờ mục tìm kiếm của GPE)

**Bài 13:UsedRange**

1./ Mc sau đây sẽ tô màu các ô công thức trong vùng sử dụng của trang tính đang kích hoạt

**Sub ColorAllFormulae()**

ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas).Interior.ColorIndex = 6

**End Sub**

2./ Vùng giao nhau với các cột cho trước & đặt tên 1 vùng dữ liệu

Ta khảo sát tiếp Mc sau:

**Sub UsedRange()**

Dim lRow As Long, bCol As Byte

2 lRow = Worksheets("S1").UsedRange.Rows.Count

3 bCol = Worksheets("S1").UsedRange.Columns.Count

With ActiveSheet

5 MsgBox Intersect(.Range("c:q"), .UsedRange).Address

End With

7 ThisWorkbook.Names.Add "Matrix", "=!r2c2:r" & lRow & "c" & bCol

'expression.Add(Name, RefersTo, Visible, McType, ShortcutKey, Category, NameLocal, \_

RefersToLocal, CategoryLocal, RefersToR1C1, RefersToR1C1Local)

**End Sub**

Dòng lệnh 2 được hiểu là số dòng chứa dữ liệu của Sheets(“S1”) đem gán vô biến lRow

Tương tự dòng lệnh 3: biến bCol sẽ chứa số cột có dữ liệu;

Dòng 5 cho ta biết địa chỉ gioa nhau giữa vùng chứa dữ liệu & các cột từ ‘C’ đến ‘Q’;

Dòng 7 các ô từ dòng 2, cột 2 đến ô cuối phải nhất được gán tên là ‘Matrix’

3./ Duyệt các ô trong 1 hàng & trong tất cả các cột của vùng chứa dữ liệu

**Sub OutputAddress()**

Dim myRange As Range, rRng As Range, cRng As Range

Dim intUnit As Integer

Dim StrR As String, StrC As String, Xh As String

Xh = Chr(10) & Chr(13)

Set myRange = ActiveSheet.UsedRange

For Each rRng In myRange.Rows

StrR = StrR & rRng.Address & Xh

For Each cRng In rRng.Cells

StrC = StrC & rRng.Address

Next

Next

MsgBox StrR, , "Row"

**End Sub**

4./ Nhân toàn bộ các ô chứa công thức số với 1 giá trị

**Sub NegativeAllNumberFormula2()**

On Error Resume Next

With Range("IV65536")

.Value = -1

.Copy

ActiveSheet.UsedRange.SpecialCells \_

(xlCellTypeFormulas, xlNumbers).PasteSpecial \_

xlPasteValues, xlPasteSpecialOperationMultiply

.Clear

End With

**End Sub**

5./ Xóa dòng theo điều kiện của 1 cột (‘D’) chứa ô trống

**Sub DeleteRowsWithSpecifiedData()**

'Looks in Column D and requires Column IV to be clean

Columns(4).EntireColumn.Insert

With Range("D1:D" & ActiveSheet.UsedRange.Rows.Count)

.FormulaR1C1 = "=IF(RC[1]="""",NA(),IF(RC[1]=""Not Needed"",NA()))"

.Value = .Value

On Error Resume Next

.SpecialCells(xlCellTypeConstants, xlErrors).EntireRow.Delete

End With

On Error GoTo 0

Columns(4).EntireColumn.Delete

**End Sub**

6./ Phóng đại vùng chứa dữ liệu khi sheet được kích hoạt

**Private Sub Worksheet\_Activate()**

Application.EnableEvents = True

Application.WindowState = xlMaximized

ActiveSheet.UsedRange.Select

ActiveWindow.Zoom = True

**End Sub**

7./ Lập danh sách địa chỉ vùng chứa dữ liệu

**Private Sub Workbook\_BeforeSave \_**

**(ByVal SaveAsUI As Boolean, Cancel As Boolean)**

'Put in the UsedRange Address of Sheet1 Book1.xls (this workbook)

Sheet2.Cells(Cells(65432).End(xlUp).Row + 1, 1) = Sheet1.UsedRange.Address

**End Sub**

**Bài 14:CurrentRegion Property (An hem sinh đôi của usedRange)**

1./ Sự khác biệt giữa CurrentRegion & UsedRange

Giả dụ chúng ta có trang tính ‘S1’ đang được kích hoạt, Tại cột A, từ A1 đến A9 & B1 đến B9 có dữ liệu đã nhập, ta thêm vài giá trị vô ô i14 & i15;

Khi đó ta chạy macro ( Mc)

**Sub UsedRange()**

Dim rTable As Range

Set rTable = Sheet1.UsedRange

MsgBox rTable.Address, , "0"

**End Sub**

Trong hộp thoại sẽ là $A$1:$I$15; Còn khi chạy Mc có nội dung

**Sub CurrentRegion1()**

Dim rTable As Range

Set rTable = Sheet1.Range("A1").CurrentRegion

MsgBox rTable.Address, , "1"

Set rTable = Sheet1.Range("i13").CurrentRegion

MsgBox rTable.Address, , "2"

With Sheet1

Set rTable = .Range(.Range("c2"), \_

.Cells(65536, .Range("IV1").End(xlToLeft).Column).End(xlUp))

End With

MsgBox rTable.Address, , "3"

**End Sub**

Sẽ xuất hiện lần lược 3 hộp thoại sau

1: $A$1:$B$9

2: $I$13:$I$15

3: $B$2:$C$9

Như vậy 3 hộp thoại cuối đưa ra 3 địa chỉ hoàn toàn khác so với vùng sử dụng (do Mc đầu tiên đưa ra). Những địa chỉ này hoàn toàn tùy thuộc vào vị trí ta đang đứng & bắt đầu gọi thực hiện CurrentRegion.

2./ Điều kì diệu của CurrentRegion

Tiếp đến ta xét đến một điều kì diệu & vô cùng thông minh của excel. Để vậy, chúng ta nhập tiếp các tên người vô cột E, bắt đầu từ E2 đến E9; Còn từ F2 đến F9 là những con số bất kỳ;

Sau đó ta cho chạy Mc sau:

**Sub TableWithHeaders()**

Dim rTable As Range: Dim lHeaderRow As Long

Set rTable = Sheet1.Range("E1").CurrentRegion

lHeaderRow = rTable.ListHeaderRows

MsgBox rTable.Address, , "A"

If lHeaderRow > 0 Then

Set rTable = rTable.Resize(rTable.Rows.Count - lHeaderRow)

MsgBox rTable.Address, , "B"

Set rTable = rTable.Offset(1)

MsgBox rTable.Address, , "C"

End If

**End Sub**

Nếu thực hiện đúng các thao tác như đã nêu, các bạn chỉ nhận được 1 hộp thoại duy nhất mang ký hiệu ‘A’, với nội dung như sau: $E$1:$F$9 (Giống trường hợp hộp thoại số (2) như trên);

Tiếp theo ta sửa nội dung của ô F2 thành chuỗi: ‘SoTien’

Lần chạy lại Mc kỳ này, ta thu thêm 2 hộp thoại mới

(B): $E$1:$F$7

(C): $E$2:$F$8

Ở đây trường hợp (C) dùng phương thức OFFSET() của một vùng nên vùng mới tăng so với vùng trước nó (chưa dùng phương thức OFFSET()) một dòng

Các vấn đề còn lại, các bạn ngẫn nghĩ & tự rút ra kết luận cho chính mình;

**What Constitutes a Heading/Header Row**

If your table is numeric data and you headings are text (or vice verca), Excel will assume **row 1 of the table as a header row**. However, if your data **AND** headings are both numeric, or both text, Excel will consider your table as having **NO headers.** The way to overcome this is to make your headings different to that of the data. This can be done via bolding, font color/size etc.

Or, should you simply know for a fact that row 1 of the table **IS a header row** you can use the code below; (Các bạn thông cảm cho vốn tiếng anh bé tẹo của mỉnh & tự đọc lấy nha!)

3./ Truy xuất từng cột dữ liệu trong vùng CurrentRegion

Để làm rõ hơn vấn đề truy xuất dữ liệu của 1 cột nào đó, chúng ta xét tiếp 1 Mc nữa, sau đây:

**Sub LoopColsSheet()**

Const Cot = 2

Dim wSh As Worksheet: Dim Rng As Range

For Each wSh In Worksheets

Select Case UCase(wSh.Name)

Case "S2", "S1"

'Do nothing

Case Else

For Each Rng In wSh.Range("A5").CurrentRegion.Columns(Cot).Cells

MsgBox Rng , , “4”

Next Rng

End Select

Next wSh

**End Sub**

Nếu ta cho Mc chạy, ta sẽ thu được thông tin dữ liệu cùa cột 2

Nếu ta thay Cot = 9, & cho chạy lại Mc, ta vẫn thu được từng ấy hộp thoại 4 mà thôi.

4./ Biến chứa vùng CurrentRegion

Tương tự như vậy, ta xét thêm trường hợp sau

**Sub Matric()**

Dim Mang, iJ As Long

Mang = Sheets("S1").Range("a1").CurrentRegion.Resize(, 3).Value

For iJ = 1 To UBound(Mang, 1)

MsgBox Mang(iJ, 2), , "5"

Next iJ

Exit Sub: **End Sub**

Trong Mc có 2 giá trị là 3 & 2; Ta chạy thử nhiều lần với các giá trị này tăng dần xem sao. Mình ngờ rằng kết quả sẽ như Mc trên nó!

5./ CurrentRegion & copy các cột dữ liệu

Ví dụ ta có dữ liệu của năm trước tại cột A:C Bắt đầu từ cột E cách đều 4 cột là dữ liệu của các tháng trong năm hiện thời; (Mỗi tháng gồm 3 cột dữ liệu & cách tháng sau nó 1 cột trống)

Nhiệm vụ đề ra là chép 12 tháng dữ liệu vô ba cột lưu dữ liệu năm trước (tại cột A:C)

Nhiệm vụ này chúng ta giao cho Mc sau:

**Sub Copy3Columns()**

Dim Rng As Range: Dim lRow As Long

Set Rng = Range("E1")

lRow = Range("A" & Rows.Count).End(xlUp).Row + 1

While Rng.Value <> ""

Rng.CurrentRegion.Copy Range("A" & lRow)

lRow = lRow + Rng.CurrentRegion.Rows.Count

Rng.Resize(, 4).EntireColumn.Delete

Set Rng = Range("E1")

Wend

Set Rng = Nothing

**End Sub**

Mc Copy3Columns có dòng lệnh 1: khai báo hai biến sẽ dùng;

D2: Ta chọn & kích hoạt ô ‘E1’

D3 : thêm 1 vô giá trị dòng cuối của dữ liệu lưu gán vô biến lRow đã khai báo ;

D4 & D9 : Thiết lập vòng lặp cho đến khi thỏa điều kiện giá trị chứa trong biến Rng là trống ;

D5 : Vùng dữ liệu lưu được chép thêm từ vùng CurrentRegion ;

D6 : Xác định lại dòng cuối của dữ liệu lưu (đã + 1)

D7 : Xóa 4 cột vừa chép ;

D8 : Xác lập lại vùng chọn

**Bài 15 :Bước đầu về phương thức Intersect**

Trong cửa sổ VBA ta gỏ Intersect , quét chọn toàn bộ và nhấn {F1} ta sẽ nhận được phần trợ giúp về phương thức Intersect như sau:

**Intersect Method**   
Returns a Range object that represents the rectangular intersection of two or more ranges.  
expression.Intersect(Arg1, Arg2, ...)  
expression Optional. An expression that returns an Application object.  
Arg1, Arg2, ... Required Range. The intersecting ranges. At least two Range objects must be specified.  
Example  
This example selects the intersection of two named ranges, rg1 and rg2, on Sheet1. If the ranges don't intersect, the example displays a message.  
Worksheets("Sheet1").Activate  
Set isect = Application.Intersect(Range("rg1"), Range("rg2"))  
If isect Is Nothing Then  
MsgBox "Ranges do not intersect"  
Else  
isect.Select  
End If

Tiếp tục ta xem thêm một số ví dự sau:  
***1./ Ví dụ khi thay đổi trị của một ô trong vùng***

**Private Sub Worksheet\_Change(ByVal Target As Range)**

StrC = "The active cell does "

If Intersect(ActiveCell, Range("A1:A9")) Is Nothing Then

MsgBox StrC & "NOT Intersect A1:A9", , Target.Address

Else

MsgBox StrC & "Intersect A1:A9", , Target.Address

End If

If Not Intersect(Target, Range("A2,B1:B9,C4:D9")) Is Nothing Then

MsgBox "Hello", , "A2,B1:B10,C5:D9"

ElseIf Not Intersect(Range(“A1:D9”,Target) Is Nothing then

MsgBox "A1:D9" ,, "Hello!"

End If

**End Sub**

***2./ Liên quan đến vùng được đặt tên:***  
  
Nếu ta đã đặt tên cho vùng nào đó trong bảng tính là "MyRang" thì khi ta đụng đến 1 ô trong vùng đó, sẽ nhận được thông báo:

**Private Sub Worksheet\_SelectionChange(ByVal Target As Range)**

Dim MyName As Name

On Error Resume Next

If Range("MyRang") Is Nothing Then Exit Sub

On Error GoTo 0

If Not Intersect(Target, Range("MyRang")) Is Nothing Then

MsgBox Range("MyRang").Name, , "Hello"

End If

**End Sub**

*3./ Tô màu nền của vùng được nhập các số ngẫu nhiên*  
  
Khi ta chọn vùng từ A7 đến A35, sau đó nhập vô thanh công thức chuỗi: =INT(19\*RAND())+32. Chúng ta kết thúc bằng tổ hợp CTRL+ENTER thì đoạn mã sau sẽ tô màu nền theo trị trong ô

**Private Sub Worksheet\_Change(ByVal Target As Range)**

Dim rgArea As Range, rgCell As Range

Dim iColor As Integer

' Get the intersect of the target & the proper range

Set Target = Intersect(Target, Range("A6:A62"))

If (Not Target Is Nothing) Then

For Each rgArea In Target.Areas

For Each rgCell In rgArea.Cells

With rgCell

If .Value < 56 Then .Interior.ColorIndex = .Value

End With

Next rgCell, rgArea

End If

Exit Sub: **End Sub**

***4./ Phương thức Union() song hành:***

Code:

**Private Sub Worksheet\_Change(ByVal Target As Excel.Range)**

Dim Rang As Range

Set Rang = Union([A1], [A3], [A5], [A7], [A9], [B1], [B3], [B5], [B7], [B9], [C1], [C3], [C5], [C7], [C9])

Set Rang = Union(Rang, [E2], [E4], [E6], [E8], [F2], [F4], [F6], [F8], [G2], [G4], [G6], [G8], [H2])

If Intersect(Target, Union(Rang, [D3], [D5])) Is Nothing Then Exit Sub

If Not Intersect(Target, Rang) Is Nothing Then

With Target.Offset(0, 1)

.Value = .Value + Target

End With

ElseIf Not Intersect(Target, [D4]) Is Nothing Then

With Range("E4")

.Value = .Value + [D4]

End With

Else

With Range("E5")

.Value = .Value + [D5]

End With

End If

**End Sub**

Đoạn code sau cho phép ta chép các hàng intersect với vùng là một số ô trong 1 cột, mà các hàng này có ô trong cột chọn không chứa giá tri:  
(Cụ thể: ta chọn vùng từ 'J3:J9' mà trong đó giá trị tại J5 & J8 = ""; thí khi chạy macro chúng ta sẽ có hai dòng dữ liệu 5 & 8 bên sheets('S2')):

Code:

**Sub CopyRows()**

Dim UniRange As Range, Rng As Range

For Each Rng In Selection

With Rng

If .Value = "" And .Offset(0, 1).Value <> "" Then

If UniRange Is Nothing Then

Set UniRange = .EntireRow

Else

Set UniRange = Application.Union(UniRange, .EntireRow)

End If: End If

End With

Next Rng ' MsgBox UniRange.Address

UniRange.Copy Destination:=Sheets("S2").Range("A65536").End(xlUp).Offset(1, 0)

Exit Sub:  **End Sub**

***5./ Một cách khác để biến các chuỗi nhập vô cột ‘D’ đều viết hoa.***

**Private Sub Worksheet\_SelectionChange(ByVal Target As Range)**

Dim Rang As Range: Dim StrC As String

Set Rang = Union([A1], [A3], [A5], [A7], [A9], [B1], [B3], [B5], [B7], [B9], [C1], [C3], [C5], [C7], [C9])

Set Rang = Union(Rang, [E2], [E4], [E6], [E8], [F2], [F4], [F6], [F8], [G2], [G4], [G6], [G8], [H2], [H4])

StrC2 = "D1:D999" ‘ !!! \*\*\* !!!

If Target.Cells.Count > 1 Or Target.HasFormula Then Exit Sub

On Error Resume Next

Application.EnableEvents = False

If Not Intersect(Target, Union(Rang, [h6], [h8], [i2], [i4], [i6], [i8])) Is Nothing Then

Target.Value = UCase(Left(Target.Value, 1)) & Mid(Target.Value, 2)

ElseIf Not Intersect(Target, Range(StrC2)) Is Nothing Then

Target.Value = UCase(Target.Value)

End If

Application.EnableEvents = True

On Error GoTo 0

**End Sub**

***6./ Một cách nhập tự động ngày hiện hành vô trường [NgThang] của CSDL***  
  
Nếu ta có CSDL mà cột B chứa [MaHg] & cột C chứa ngày nhập, cột F chứa ngày xuất thì đoạn mã sau sẽ cho phép tự động nhập ngày hiện hành khi ta nhập vô cột trước nó là mã vật tư, hàng hoá nhập hay xuất.

**Private Sub Worksheet\_Change(ByVal Target As Range)**

If Not Intersect(Target, Range("B:B,E:E")) Is Nothing Then

If Not IsEmpty(Target) Then

Target.Offset(0, 1).Value = Date

Else

Target.Offset(0, 1).Value = Empty

End If

End If

**End Sub**

***7./ Phương thức Intersect và copy vùng dữ liệu***  
  
Dùng phương thức Intersect để kiểm tra vùng dữ liệu khi cần copy   
**Trường hợp A:**

Code:

**Sub ShtChange()**

Dim Rang0 As Range, IntersectRng As Range

Sheets("S4").Select: Set Rang0 = Range("J3:J16")

Set IntersectRng = Intersect(Selection, Rang0)

If IntersectRng Is Nothing Then Exit Sub

CopyC IntersectRng

**End Sub**

**Private Sub CopyC(Target As Range)**

Dim lRow As Long

Application.ScreenUpdating = False: Application.EnableEvents = 0

lRow = Target.Row ' lRow = IntersectRng.Row

'Xoá 1 dòng chứa dữ liệu

Range("A" & lRow & ":J" & lRow).ClearContents

Range("A" & lRow + 1 & ":I16").Copy '& Copy Các dòng dưới lên thay chỗ

Range("A" & lRow & ":I15").PasteSpecial xlPasteValues

Range("A16:I16").ClearContents

Application.EnableEvents = True: Application.ScreenUpdating = -1

**End Sub**

**Trường hợp B**

Code:

**Sub CopyCol()**

Dim iRow As Integer

iRow = ActiveCell.Row

If Not Intersect(ActiveCell, Range("$C$3:$M$3000")) Is Nothing Then

Application.EnableEvents = False

If Range("B" & iRow) = "" Then

Range("O" & iRow) = "P"

Else

Range("B" & iRow).Select

Selection.Copy

Range("Q7").Select

If Not Range("Q7") = "" Then

Range("Q6:Q150").Select

Selection.End(xlDown).Offset(1, 0).Select

'Selection.Offset(1, 0).Select

End If

Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, \_

SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Range("B" & iRow) = "": Range("O" & iRow) = "P"

End If

Else

Range("O" & iRow).Interior.ColorIndex = xlColorIndexNone

End If

Application.EnableEvents = True

**End Sub**

**Trường hợp C**

Code:

**Sub Copy\_Stuff()**

On Error GoTo ExitSub

Dim rCols As Range, CheckCells As Range, rCell As Range, CopyRange As Range

Dim bDem As Long

Set rCols = Sheet1.Range("A:I, L:R")

Set CheckCells = Intersect(Sheet1.Range("L9:L65536"), Sheet1.UsedRange)

For Each rCell In CheckCells

If rCell.Value > 0 Then

bDem = bDem + 1

If bDem = 1 Then

Set CopyRange = rCell.EntireRow

Else

Set CopyRange = Union(CopyRange, rCell.EntireRow)

End If

End If

Next rCell

Set CopyRange = Intersect(rCols, CopyRange)

For Each rCell In CopyRange.Areas

bDem = Sheet3.Rows.Count

Set rCols = Sheet2.Cells(bDem, rCell.Column).End(xlUp).Offset(1, 0)

rCell.Copy rCols

Next rCell

ExitSub:  **End Sub**

**Trường hợp D**

Code:

**Sub Get\_Data()**

On Error Resume Next

Dim vRange As Range, MatchCells As Range, cell As Range

Dim NgDau As Date, NgCuoi As Date

Sheets("S2").Select

NgDau = #9/16/2006#: NgCuoi = 9 + NgDau

Set vRange = Range("D:D", "F:F")

For Each cell In vRange.SpecialCells(xlConstants, xlNumbers)

If cell.Value >= NgDau And cell.Value <= NgCuoi Then

If MatchCells Is Nothing Then

Set MatchCells = cell.EntireRow

Else

Set MatchCells = Union(MatchCells, cell.EntireRow)

End If

End If

Next cell

Intersect(MatchCells, vRange).Copy

Sheets("S3").Range("G2").PasteSpecial (xlValues)

Application.CutCopyMode = False: Sheet2.Select

MsgBox Intersect(MatchCells, vRange).Address, , MatchCells.Address

Selection.Offset(0, 1).SpecialCells(xlConstants, xlTextValues).ClearContents

**End Sub**

Trường hợp E

Code:

**Sub GenerateSht()**

'The following will generate the Over/Under worksheet

'Range copied will depend on whether or not the value in column 11 is greater than 0

'Columns 9 and 10 are not to be copied.

Dim Cols As Range, SearchCell As Range, CopyRange As Range

Dim wSheet As Object, LastRow As Long, lDem As Long

With Application

.ScreenUpdating = False: .Calculation = xlCalculationManual

End With

Set wSheet = Sheets("DNgh"): wSheet.Range("A9:T999").ClearContents

Sheets("S4").Select

Set Cols = Range("A:I, L:S"): Set SearchCell = Range("B9")

Set wSheet = Sheets("S4")

With wSheet

Do While Len(SearchCell) > 1 'Bat Dau Tu B9 den khi Len(B(i)) > 1

If SearchCell.Offset(0, 10) <> 0 Then

lDem = lDem + 1

If lDem = 1 Then

Set CopyRange = SearchCell.EntireRow

Else

Set CopyRange = Union(CopyRange, SearchCell.EntireRow)

End If

End If

Set SearchCell = SearchCell.Offset(1, 0) 'Set SearchCell Variable for next loop

Loop

End With

Set CopyRange = Intersect(Cols, CopyRange)

Set wSheet = Sheets("DNgh") 'Release

For Each SearchCell In CopyRange.Areas

'insert the copied values on the DNgh sheet

lDem = wSheet.Rows.Count

Set Cols = wSheet.Cells(lDem, SearchCell.Column).End(xlUp).Offset(1, 0)

SearchCell.Copy: Cols.PasteSpecial (xlValues)

Next SearchCell

With wSheet 'Inserts the underscore in the empty cells J & K of the DNgh sheet

LastRow = .Range("B" & Rows.Count).End(xlUp).Row 'Set variable to the specified range

.Range("B9:B" & LastRow).Offset(0, 8).Resize(, 2) = "\_\_\_\_\_\_\_\_\_"

End With

Set wSheet = Nothing 'Release the variable from memory

With Application

.ScreenUpdating = True: .Calculation = xlCalculationAutomatic

End With

**End Sub** 'END GenerateSht

**(Phần đọc thêm: Đây là phần dịch của bạn sinh viên trong phần trợ giúp về đối tượng/phần tử Application )**

Application Object

Phần tử Application chính là bản thân excel. Nó là bản chất của mọi phần tử trong excel. Phần tử Application bao gồm:  
  
• Application-wide settings and options (many of the options in the Options dialog box (Tools menu), for example).  
  
• Methods that return top-level objects, such as ActiveCell, ActiveSheet, and so on.  
  
• Các khả năng thiết lập và tùy chọn thông số phong phú cho phần tử (phần lớn các tùy chọn ở trong hộp thoại Options (menu Tools))  
• Các phương thức để chỉ định các đối tượng hiện hành, như ActiveCell, ActiveSheet, v.v.  
  
Using the Application Object  
  
Thao tác với phần tử Application  
  
Use the Application property to return the Application object. The following example applies the Windows property to the Application object.  
  
Sử dụng các thuộc tính của phần tử Application để thao tác với đối tượng Application. Trong ví dụ sau ta thao tác với thuộc tính Windows của Application.  
  
Application.Windows("book1.xls").Activate  
  
The following example creates a Microsoft Excel workbook object in another application and then opens a workbook in Microsoft Excel.  
  
Còn trong ví dụ sau đây ta tạo ra một đối tượng workbook của Microsoft Excel   
  
Set xl = CreateObject("Excel.Sheet")  
xl.Application.Workbooks.Open "newbook.xls"  
  
Remarks  
  
Lưu ý  
  
  
Many of the properties and methods that return the most common user-interface objects, such as the active cell (ActiveCell property), can be used without the Application object qualifier. For example, instead of writing Application.ActiveCell.Font.Bold = True, you can write ActiveCell.Font.Bold = True.  
  
Có nhiều thuộc tính và phương thức chỉ định đến các đối tượng thường dùng, như đến cell hiện hành (thuộc tính ActiveCell), các thuộc tính này có thể dùng mà không cần từ khóa Application. Ví dụ: thay vì viết Application.ActiveCell.Font.Bold = True, ta có thể viết ActiveCell.Font.Bold = True.

**Bài 16:TÌM HIỂU VỀ PHẦN TỬ APPLICATION**

Trong các macro trước đây, chúng ta đã thấy các câu lệnh đề cập đến phần tử Application, ví dụ :

PHP Code:

1 Application.ScreenUpdating = False   
2 Vloop = Application.VLookup(List, Rng, 4, False)    
3 Set myRange = Worksheets("Sheet1").Range("A1:C10")   
4 answer = Application.WorksheetFunction.Min(myRange)   
MsgBox answer

Phần từ Application chính là bản thân microsoft excel. Nó là bao trùm, là vũ trụ của mọi phần tử trong excel.  
  
Nếu chúng ta đếm các tính chất & phương pháp của phần tử này thì nhiều vô kể. Để tìm hiểu các tính chất & phương thức của nó, ta có tối thiểu 2 cách sau:  
  
**1./ Vô CS (cửa sổ) Object Browser**.  
  
**2./ Đơn giản hơn**, sau khi vô CS VBE, ta nhập từ “Application.”, sau khi ta nhập dấu chấm, sẽ có danh sách đổ xuống, ghi hầu hết các phương thức & tính chất của vũ trụ excel này.  
  
Bỡi lẽ do nhiều quá những phương thức & tính chất, nên có một số người đã phân loại chúng theo các nhóm sau :  
  
**1\* Các tính chất & phương thức toàn cục (global)** : Đó là trên 40 tinh chất & phương pháp dùng trực tiếp, không cần dùng từ Application ở trước. Chúng ta sẽ nói kỹ nhóm này ở phía dưới ;  
  
**2\* Các hàm của trang tính**; Nếu một ai chưa biết các hàm này, thì chưa thể gọi là đã biết về excel; Chúng ta xem lại các dòng lệnh 2 & 4 nêu trên. Tại dòng lệnh 2, biến Vloop sẽ chứa kết quả trả về của hàm VLOOPKUP(). Dòng lệnh 4, biến answer sẽ chứa giá trị thấp nhất trong mãng giá trị được chúng ta gán tên là myRange   
  
**3\* Các thiết chế cho môi trường làm việc**: Bạn có thể nhấn {F9} để tính toán lại bảng tính hiện hành

PHP Code:

5  Worksheets("Sheet1").UsedRange.Columns("A:C").Calculate   
6  Application.Calculation = xlCalculationManual

(Nếu ta cần diễn dịch dòng lệnh 5, thì nên đọc từ phải qua trái, như sau:   
  
Tính lại cột A đến cột C của vùng sử dụng trên trang tính có tên ‘Sheet1’  
Cách dễ tiếp thu hết các lệnh loại này là ta dùng bộ thu macro để biết một số hành động của chúng ta tác động lên trang tính được viết bỡi ngôn ngữ VBA là như thế nào  
  
**4\* Các thông tin về môi trường làm việc**. Excel có khoảng hơn 30 tính chất như vậy. Chúng cho ta biết các thông tin như Bạn đang làm việc trong windows hay không, bạn còn bao nhiêu bộ nhớ khả dụng. . .  
Ví dụ:

PHP Code:

MsgBox "Welcome to Microsoft Excel version " & \_    
    Application.Version & " running on " &  Application.OperatingSystem & "!"   
  
ActiveWorkbook.Windows(1).Caption = "Consolidated Balance Sheet"   
ActiveWorkbook.Windows("Consolidated Balance Sheet") \_   
    .ActiveSheet.Calculate

**5\* Các thiết định về macro**: Ví dụ như câu lệnh trên cùng của bài, dùng để không thay đổi màn hình khi macro chạy. Điều này rất cần thiết & không nên quên khi ta phải duyệt một số lượng lớn các record (Tìm mã mặt hàng, họ tên học sinh trong một cơ sở dữ liệu lớn. . .). Thêm thông báo ở dòng trạng thái trong macro sau

PHP Code:

Sub Statusbar()   
 Dim oldStatusBar, StrC As String   
 Dim iDem As Integer, iZ As Long, jW As Long   
    
 oldStatusBar = Application.DisplayStatusBar   
 Application.DisplayStatusBar = True   
 StrC = "\*\*\*\*\*\*\*\*\*\*\*\*"   
 For iZ = 1 To 60000   
    Cells(iZ, 1) = iZ   
    For jW = 1 To 10 ^ 4   
    Next jW   
    If iZ Mod 6000 = 0 Then   
        iDem = iDem + 1   
        Application.Statusbar = Left(Left(StrC, iDem) & "..........", 10)   
    End If   
 Next iZ   
    Application.Statusbar = False   
 Application.DisplayStatusBar = oldStatusBar   
End Sub

**6\* Những tính chất & phương thức khó xếp vô một nhóm nào**. Như đổi đơn vị đo lường, gởi thư điện từ, gọi hệ trợ giúp . . .  
Ví dụ:

PHP Code:

 Application.Help "OTISAPP.HLP", 65527

Trong cú pháp lệnh này, gồm 2 đối số tùy chọn; Đối số đầu là tên file trợ giúp; đối số sau là chỉ mục của file. Nếu ta không đưa ra chỉ mục thì hộp thoại trợ giúp của excel sẽ hiện ra!

PHP Code:

 Worksheets("Sheet1").PageSetup.LeftMargin = \_   
        Application.InchesToPoints(2.5)   
Worksheets("Sheet1").PageSetup.LeftMargin = \_   
        Application.CentimetersToPoints(5)

**Bài 17:CÁC TÍNH CHẤT & PHƯƠNG PHÁP TOÀN CỤC.**

Trong các câu lệnh macro, có những từ có thể đứng một mình, trong khi đó cũng có những từ cần phải đứng sau một từ khác chỉ rõ tập hợp bố mẹ của chúng; Ví dụ:

PHP Code:

MsgBox Cells.Item(9).Address, , Range("B1:C9").Item(3).Address

Trong ví dụ trên tập hợp cha mẹ, trong trường hợp thứ nhất là toàn bộ các ô trên trang tính; trong trường hợp thứ hai là 18 ô trên trang tính nào đó đang được kích hoạt  
  
Worksheets("Sheet1").Cells.ClearContents  
Câu lệnh này dùng để xóa toàn bộ nội dung các ô trong trang tính được nêu tên.  
  
Tương tự như trên, các từ Boder hay Interior, . . . cũng phải đi kèm với cha, mẹ cũa chúng; không.thể đứng riêng một mình được.  
Nhưng cũng có một số từ có thể đứng riêng, như Selection, WorkBooks, . . .   
  
Sao lại như vậy?  
  
Chúng ta phải liên tưởng đế các câu sau:  
“ Ông Chủ Nhà Trắng đã tuyên . . . .”  
“Ngày mai tôi sẽ bay từ Hà nội đến Pari. . .”  
Ở câu đầu, ta biết ngay, đó là tổng thống Mĩ; Ở câu hai, người nghe sẽ biết ngay là anh/chị này đã có hộ chiếu đến Pháp.   
Ta xét trường hợp Selection, như sau:

PHP Code:

Sub DongTrongXuongDuoi()   
'Assumes the list has a heading'   
 On Error Resume Next   
    With Selection   
        .Sort Key1:=.Cells(2, 3), Order1:=xlAscending, \_   
            Header:=xlYes, OrderCustom:=1, MatchCase:=False, \_   
            Orientation:=xlTopToBottom   
    End With   
End Sub

Với trường hợp workbooks, ta có ví dụ sau:

PHP Code:

Sub workbooks\_()   
Dim Wh As Workbook   
  
For Each Wh In Workbooks   
    If Wh.Name <> ThisWorkbook.Name Then   
        Wh.Close savechanges:=True   
    End If   
Next Wh   
End Sub

Ở trường hợp sau cùng, macro sẽ đóng tất cả các workbooks đang mở, chỉ trừ workbook có chứa macro. Và ta được biết thêm một số điều rằng,   
  
A\*) Phương thức colse có thêm đối số cho nó.   
Và nếu không có đối số, như trong trường hợp  
Workbooks.Close, thì khi excel gặp câu lệnh này, excel sẽ đưa ra cho ta hộp thoại, để hỏi chúng ta có lưu những tác động thay đổi của chúng ta lên các trang tính hay không? Nếu chương trình chúng ta chỉ là một mẫu cỏn con, chúng ta còn có thể nhớ & biết tại sao excel lại làm vậy. Nhưng nếu chúng ta có một macro đồ sộ. Khi đó chúng ta không thể biết tại sao excel lại hỏi. Để khỏi bị bất ngờ trong những trường hợp như vậy, chúng ta nên chọn cách an toàn nhất (cho chúng ta cũng như cho những người sử dụng sau này!)  
  
B\*) Phương thức Close còn tác động lên các cửa sổ. Điều này giành cho các bạn nghiên cứu thông qua phần trợ giúp của CS VBE trong excel.   
Nếu chúng ta coi phần tử Application là quả đất của chúng ta, thì Workbooks như một nước Nga, hay nước Mĩ nào đó; & Selection như nước Viết nam. Ở đây chúng ta phải hiểu ngầm rằng, ngày mai anh bay từ Tokyô đến đến Hoa thịnh đốn; cũng là nói ngày mai anh ấy sẽ bay từ Nhật bản sang Mĩ vậy. Nói đến workbooks là ta phải ngầm định ngay nó là tập hợp con ngay kế với tập hợp Application.

Không biết phải vì tầm quan trọng, hay vì cái gì khác, mà một số người lại phân nhóm nhỏ cho các tính chất & phương pháp toàn cục này:  
  
1\*/ Các tính chất để chỉ định: Selection,ThisWorkbook & các tính chất ‘Active’, như ActiveCell, ActiveSheet, ActiveWindow, ActiveWorkBook ActiveChart, ActiveDialog, ActiveMenuBar, ActivePrinter,. . ..(Cần nói rõ rằng, tính chất ThisWorkbook sẽ chỉ đến bảng tính có chứa macro, chứ nó không yêu cầu bảng tính này đang được kích hoạt hay không)  
  
2\*/ Các tập hợp vùng, các phương pháp phổ biến nhất trả về phần tử vùng, như Range, Union, Intersect, Cells, Rows & Columns, . . .  
Nhóm các phương pháp & thuộc tính này chúng ta đã đề cập tại bài đầu.  
  
3\*/ Các tập hợp trang: Các phương pháp hay tính chất trả về các tập hợp trang khác nhau trong bảng tính, như Sheets, Worksheets, Charts, DialogSheets, Modules, Excel4MacroSheets, Excel4IntlMacroSheets, . .   
  
4\*/ CÁc tập hợp khác, như AddInt, MenuBars, Names, ShortcutMenus, Toolbars, Windows & workbooks, . . .  
  
5\*/ Các lệnh, cũng như các phương pháp dùng để thực hiện các việc thông thường, như Calculate, Evaluate, Run & Sndkeys; Cùng với các lệnh dùng cho DDE (liên kết dữ liệu động). Cái này quả là cao cấp & xa xỉ đối với nhiều người, trong đó có tôi!  
Để kết luận, xin được nhắc lại là, những tính chất & phương pháp toàn cục này, tuy không đi kè với Application, nhưng đó là bố đẽ ra chúng.

**Bài 18:ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TỰ NHAU TRONG MACRO.**

Giả sử nhiệm vụ đề ra là ta fải viết macro để theo dõi việc cung cấp hàng hóa từ nhà cung cấp như bảng được trích ra như sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ngay | NhaCC | Ma | Ten | GiaMua | SoLg | Ttien |  |
|  | 1/9/2010 | Thơm | ANL | Áo NLớn | 45.000 | 5 | 225.000 |  |
|  | 1/9/2010 | Thà | ANL | Áo NLớn | 50.000 | 10 | 500.000 |  |
|  | 2/9/2010 | Hoa | ATE | Áo Trẻ em | 30.000 | 5 | 150.000 |  |
|  | 3/9/2010 | My | QNL | Quần NLớn | 60.000 | 5 | 300.000 |  |
|  | 3/9/2010 | Thơm | QNL | Quần NLớn | 70.000 | 5 | 350.000 |  |
|  | 3/9/2010 | Thà | ANL | Áo NLớn | 50.000 | 1 | 50.000 |  |
|  |  | . . . |  | . . |  |  | . . . |  |

Cụ thể nhiệm vụ là cần biết trong 1 ngày nào đó hay cả tháng ông A hay bà C đã cung cấp bao nhiêu mặt hàng, hoặc ngày nào đó bất kỳ có bao nhiêu nhà cung cấp đã nhập hàng.  
Kết quả thống kê này sẽ cho ra trang tính có tên ‘Phieu’ (trong file đính kèm).  
Mà tại trang tính này người ta đã thiết lập 2 combobox tại [C4] & [H4] để lực chọn nhà cung ứng hay ngày tháng cụ thể nào đó cần khảo sát.  
Nếu tại [C4] để trống, bảng kết quả thể hiện toàn thể các nhà cung cấp trong ngày;  
Nếu [H4] để trống, bảng kết quả sẽ là toàn bộ số hàng – tiền của người đó đã cung cấp trong tháng.  
  
Macro đó có nội dung như sau:

PHP Code:

Option Explicit   
 Dim Sh As Worksheet, Rng As Range, sRng As Range   
    
Private Sub Worksheet\_Change(ByVal Target As Range)   
 Dim NCC As Boolean, Ngay As Boolean   
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'   
    
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'   
 If Not Intersect(Target, [c4]) Is Nothing Then  'Nha CC'   
   [B7].Resize(14, 7).ClearContents   
   If [c4].Value = "" Then Exit Sub   
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))   
   If [H4].Value = "" Then Ngay = True   
   Set sRng = Rng.Find([c4].Value, , xlFormulas, xlWhole)   
2 Rem  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay   
  '     \*     \*     \*     \*     \*'   
  [B7].Resize(99).EntireRow.Hidden = False   
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'   
   Do   
      With [b21].End(xlUp).Offset(1)   
         If Ngay Then  'All\_ '   
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'   
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value   
         Else   
            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'   
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value   
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value   
            End If   
         End If   
      End With   
      Set sRng = Rng.FindNext(sRng)   
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd   
    
   With [B99].End(xlUp)   
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2   
   End With   
   Range("B99:B" & Offs).EntireRow.Hidden = True   
   Set Sh = Nothing   
'     \*     \*     \*     \*     \*'   
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'   
   [B7].Resize(14, 7).ClearContents   
   If [H4].Value = "" Then Exit Sub   
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))   
   If [c4].Value = "" Then NCC = True   
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole)   
3 Rem   If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1   
'     \*     \*     \*     \*     \*'   
   [B7].Resize(99).EntireRow.Hidden = False   
   MyAdd = sRng.Address:                        '     Application.ScreenUpdating = False'   
   Do   
      With [b21].End(xlUp).Offset(1)   
         If NCC Then                'All\_'   
            .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value '1=>Add1'   
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value   
         Else   
            If sRng.Offset(, 1).Value = [c4].Value Then ' Offs=>1; Cls=> [C4]'   
               .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value   
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value   
            End If   
         End If   
      End With   
      Set sRng = Rng.FindNext(sRng)   
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd   
   With [B99].End(xlUp)   
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2   
   End With   
   Range("B99:B" & Offs).EntireRow.Hidden = True   
   Set Sh = Nothing   
 '     \*     \*     \*     \*     \*'   
 End If   
End Sub

Macro này đã đáp ứng được yêu cầu nhiệm vụ đề ra. (Các bạn có thể trãi nghiệm qua file đính kèm để khẳng định điều đó).  
Vậy là đòi hỏi đã được đáp ứng 1 cách thỏa đáng.  
Nhưng với chúng ta, những người sản sinh ra những macro đại loại như trên thì khoang vội thỏa mãn.  
Quan sát kỹ lưỡng một chút, ta sẽ thấy macro chạy cho 1 trong 2 ComBo 1 cách độc lập.  
Có nghĩa là ta đụng vào combo nào, thì chỉ những dòng lệnh liên quan đến nó mới lay động mà thôi.  
Hơn nữa, số các dòng lệnh trong mỗi fân nhóm là như nhau.  
Từ hai nhận xét trên, ta tự đề ra nhiệm vụ đặt thừa số chung cho macro này (theo nghĩa đen của toán học đại số)  
Nôm na là ta có biểu thức 2 \* Aa + Bb \* Aa   
Thì ta làm gọn lại như Aa \* (2 + Bb) vậy mà.  
(Có nghĩa là thay vì fải dùng đến 4 fép toán đại số mới ra kết quả thì ta chỉ xài 3 fép thôi  
  
(Tôi xin dừng bài này ở đây vài ngày để các bạn nào quan tâm thử sức xem sao. Các bạn xem kỹ trong file sẽ thấy đáp án)  
  
Hẹn bài sau sẽ fân tích kỹ thêm, cách mà ta gộp thừa số chung trong macro

.

**Bài 19:ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TƯ NHAU TRONG MACRO. (Bài 2)**

.

Quan sát nội dung macro ở bài 1 bên trên, chúng ta dễ nhận ra 4 dòng không chứa mã lệnh nào; chúng chỉ chứa 5 dấu sao (\*) ở mỗi dòng.  
Nếu xem xét kỹ hơn, ta sẽ thấy các dòng lệnh giữa dòng (\*) 1 & dòng (\*) 2 cũng gần bằng với số dòng lệnh giữa dòng (\*) 3 & 4;  
Không những thế, chúng còn có nội dung rất giống nhau trong từng dòng.  
  
Nhiệm vụ bài này sẽ là: Chúng ta sẽ viết 1 macro thay cho 2 đoạn mã lệnh mà chúng ta vừa đề cập trên.  
Công việc này chưa hẵn mang í nghĩa tăng tốc cho toàn bộ chương trình, ý nghĩa của việc làm này là ở chỗ: Tôi cùng các bạn ta sẽ tiến hành thực hiện cẩn trọng từng bước để đạt múc đích làm cho macro của chúng ta dễ sử dụng & bảo trì trong tương lai  
  
Các bước đó là:  
**Bước 1./**  
Chép fân đoạn I ra 1 macro riêng (tạm gọi là macro con) & chỉnh sửa để khi ta thêm dòng lệnh vô macro cha gọi macro con, thì kết quả vẫn như ta thực hiện macro cha khi chưa cho con ở riêng, theo kiểu

PHP Code:

Sub MacroBĐ()  
  ‘ . . . . . . . . . ‘  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần I)  
‘    \*    \*    \*    \*    \*’  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần II)  
‘    \*    \*    \*    \*    \*’  
‘. . . . . . . . . .’  
End Sub

Trở thành:

PHP Code:

Sub MacroCha()  
  ‘ . . . . . . . . . ‘  
‘    \*    \*    \*    \*    \*’  
‘     (Các câu lệnh fần I) ‘  
‘    \*    \*    \*    \*    \*’  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần II)  
‘    \*    \*    \*    \*    \*’  
‘. . . . . . . . . .’  
End Sub

Code:

Sub MacroCon()

(Chứa các câu lệnh fần I) ‘<=|’

End Sub

Như vậy bước I ta thực hiện 2 nhiệm vụ:  
  
**a./ Chép đoạn mã của fần I sang macro mới:**  
  
(Macro con này chúng ta sẽ cho nó cái tên là GPE\_Copy nha các bạn)  
  
Đấu tiên ta đưa con trỏ xuống dòng trống cuối cùng của trang màn hình đang chứa macro cha

PHP Code:

Private Sub Worksheet\_Change(ByVal Target As Range)  
 ‘ .. . . . ‘  
 End Sub

(dưới dòng End Sub)  
  
Ta gỏ nhập “Sub GPE\_Copy” & {ENTER} để ta có được sau sự bổ sung của VBE như sau:

PHP Code:

 Sub GPE\_Copy()  
  
End Sub

Sau đó ta chỉ việc dùng chuột tô chọn khối các dòng lện fần I của macro cha & chép vô giữa 2 dòng lệnh vừa xuất hiện  
Sau các thao tác đó, chúng ta có kết quả như sau

PHP Code:

 Sub GPE\_Copy()  
11  [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'  
13   Do  
      With [b21].End(xlUp).Offset(1)  
15         If Ngay Then  'All\_ '  
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'  
17            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
         Else  
19            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'  
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value  
111               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
            End If  
113        End If  
      End With  
115      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
   
117   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
119   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
121   Set Sh = Nothing  
End Sub

Đó là xong nhiệm vụ chép thôi; còn nhiệm vụ chỉnh sửa nữa mới là fần quan trọng.  
Để tiện trong bước tiếp theo, tôi đã đánh số các dòng lệnh như các bạn thấy bên trên  
  
**b./ Chỉnh sửa macro con.**  
  
(Nếu đến thời điểm này mà ta :  
- Thêm dòng lệnh gọi macro con này từ ngay trước fần I macro cha  
- Vô hiệu hóa toàn bộ các dòng lệnh fần I của macro cha  
và tiến hành thao tác để chạy macro cha ( Cụ thể thay đổi [C4] ở trang tính ‘Phieu’) ta sẽ bị báo lỗi.)  
  
Chuyện đó là đương nhiện theo các lẽ sau:  
  
Ta chưa khai báo biến nào cho macro con (Vì trên cùng ta có dòng Option Explicit & dòng lệnh 12 ta xài biến MyAdd chưa khai báo chẳng hạn)  
  
Thêm nữa, trong dòng lệnh 16 ta có xài biến kiểu Range sRng, mà trên chương trình cha ta đã khai báo. & gán cho nó là ô chứa kết quả tìm được trong biến Rng.  
Như vậy ta có thể đưa khai báo sRng này làm tài sản chung cho 2 macro cha & con.  
  
Vấn đề nữa, đó là tại dòng 15 ta ra điều kiện chứa trong biến Ngay kiểu Yes/No. Như vậy chương trình con cần fải biết tại thời điểm đó biến Ngay là Yes hay No để mà ứng xử cho đúng. Có nghĩa là tham biến này fải được truyền từ chương trình cha sang cho chương trình con.  
  
Tóm lại: Ta cần rà soát từng dòng lệnh để:  
(\*) Khai báo biến dùng chung trong 1 số trường hợp  
(\*) Truyền tham biến từ chương trình cha cho chương trình con, để anh con này làm được đúng yêu cầu;  
(\*) Khai báo thêm 1 số biến fát sinh, nếu cần   
  
Ta tiến hành kiểm theo từng dòng lệnh một:  
  
D11: Không fát sinh, vì ta đang thao tác trên trang tính hiện hành;  
D12 Biến MyAdd nên là 1 biến cha & con cùng dùng (Có nghĩa là ta sẽ fải khai báo 1 biến dùng chung trong toàn module);  
(Hai dòng 13 & 14) không có vấn đề gì. . . - cho qua)  
D15, Tham biến Ngay cần được truyền từ cha sang con; Nhưng đến chương trình con nó có thể mang tiên khác chút; đó là tôi dùng tiên All\_ trong chương trình con;  
D16 & D17: Tham biến sRng xài chung như MyAdd;  
D19 Ô [H4] hiện chưa có ván đề; Hẹn bài sau sẽ fát sinh vấn đề;  
Từ đây cho đến dòng 117 sẽ là 1 trong những trường hợp ta đã xét; Các bạn tự xuy luận giúp.  
D118: Biến Offs ta có thể khai báo dùng chung, nhưng tiết kiệm tài nguyên 1 chút, ta sẽ khai báo thêm tại chương trình con.   
  
Với tất cả những gì chúng ta vừa lập luận, ta đẻ ra 2 macro cha & con có những nội dung như sau:

PHP Code:

Option Explicit  
 Dim Sh As Worksheet, Rng As Range, sRng As Range  
   
Private Sub Worksheet\_Change(ByVal Target As Range)  
 Dim NCC As Boolean, Ngay As Boolean  
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'  
   
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'  
 If Not Intersect(Target, [c4]) Is Nothing Then  'Nha CC'  
   [B7].Resize(14, 7).ClearContents  
   If [c4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))  
   If [H4].Value = "" Then Ngay = True  
   Set sRng = Rng.Find([c4].Value, , xlFormulas, xlWhole)  
2 Rem  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay  
  '     \*     \*     \*     \*     \*'  
  [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'  
   Do  
      With [b21].End(xlUp).Offset(1)  
         If Ngay Then  'All\_ '  
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'  
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
         Else  
            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'  
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value  
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
            End If  
         End If  
      End With  
      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
   
   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
   Set Sh = Nothing  
'     \*     \*     \*     \*     \*'  
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'  
   [B7].Resize(14, 7).ClearContents  
   If [H4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))  
   If [c4].Value = "" Then NCC = True  
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole)  
3 Rem   If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1  
'     \*     \*     \*     \*     \*'  
31   [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address:                        '     Application.ScreenUpdating = False'  
33   Do  
      With [b21].End(xlUp).Offset(1)  
35         If NCC Then                'All\_'  
            .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value '1=>Add1'  
37            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value  
         Else  
39            If sRng.Offset(, 1).Value = [c4].Value Then ' Offs=>1; Cls=> [C4]'  
               .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value  
301               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value  
            End If  
303         End If  
      End With  
305      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
307   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
309   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
311   Set Sh = Nothing  
 '     \*     \*     \*     \*     \*'  
 End If  
End Sub

**(Bổ sung số cho các dòng lệnh từ 31. . 311 ngày 19/09/2010)**

Code:

Sub GPE\_Copy(Cls As Range, All\_ As Boolean, Optional Offs As Integer = -1, Optional Add1 As Byte)

Dim MyAdd As String

[B7].Resize(99).EntireRow.Hidden = False

MyAdd = sRng.Address: Application.ScreenUpdating = False

Do

With [b21].End(xlUp).Offset(1)

If All\_ Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

Else

If sRng.Offset(, Offs).Value = Cls.Value Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

End If

End If

End With

Set sRng = Rng.FindNext(sRng)

Loop While Not sRng Is Nothing And sRng.Address <> MyAdd

With [B99].End(xlUp)

If .Row < 13 Then Offs = 15 Else Offs = .Row + 2

End With

Range("B99:B" & Offs).EntireRow.Hidden = True

Set Sh = Nothing

End Sub

Ở đây ta dùng tham số Cls thay cho [H4]; & ta bỏ từ khóa ‘Rem’ tại dòng lệnh mang số 2 là & thêm dòng lệnh

PHP Code:

Exit Sub

kề dưới dòng này nữa (Dòng lệnh ta vừa bỏ chữ Rem) là mọi chuyện trở nên mĩ mãn.  
  
Bây giờ ta có thể chạy macro cha; Đến lúc cần, macro cha sẽ gọi macro con chạy tiếp sức cho đến kết quả khi ta chọn khách hàng trong ô [C4]  
Còn nếu ta chọn ngày nhập hàng tại [H4] Chỉ có macro cha chạy 1 mạcch cho đến kết quả cuối cùng mà thôi.  
  
(Sẽ có bạn thắc mắc: Làm chio cho fức tạp vậy? Như cũ đã không tốt rối sao!")  
  
Câu nghi vấn này sẽ được trả lời ở những bài sau.

**Bài 20:ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TƯ NHAU TRONG MACRO. (Bài 3)**

Ở bài 2, chúng ta ta viết lại macro con, mà thực tế nó được chép từ khối I của macro cha. Lúc đó macro con sẽ được sửa lại để làm nhiệm vụ thay cho khối lệnh I đã bị vô hiệu hóa sau đó.  
  
Nhiệm vụ của ta bài này sẽ là biến macro con này làm fần việc của khối lệnh II (Nằm giữa 2 dòng chứa 5 dấu sao 3 & 4 mà ta đã đề cập ở bài 2 nêu trên.)  
  
Để tiện cùng nhau trao đổ thông tin, tôi xin fép đánh số các dòng lệnh ở khối II của macro cha ( đoạn mã gần cuối bài 2) & (các dòng lệnh của macro con đã được gán số như fần đầu bài 2). Hơn nữa, các dòng lệnh trong macro con đều bắt đầu từ số 1; Còn các dòng lệnh trong khối lệnh II ta đã bắt đầu bỡi số 3.  
  
Bây giờ ta cần đối chiếu từng cặp dòng lệnh giữa 2 khối này.; Nhưng để tiết kiệm thời gian cũng như có dịp đi sâu vô các dòng lệnh khác biệt nhau, cho fép tôi không nói lại những dòng lệnh gần giống hệt nhau cả về cú fáp cũng như kết quả thực hiện chúng.  
  
Ta đi vô dòng 35 có nội dung: If NCC Then (& tương ứng với nó là dòng 14 If Ngay Then)  
  
Như cuối bài 2 ta thấy, khi gọi macro con, chúng ta đã truyền cho nó tham số Ngay  
  
Thì dòng lệnh gọi macro con thay cho khối II cũng sẽ truyền cho macro con tham biến NCC;   
Nhưng khi đến macro con này, nó có thể mang 1 tên khác, chẳng hạn All\_ như bạn sẽ thấy. Tất nhiên, nó vẫn có thể giữ nguyên tên là Ngay, nhưng như vậy ta tự làm khó ta vì dễ làm ta tẩu hỏa nhập ma lúc nào đó & lộn tùng xèn không đáng có.  
  
Trong các dòng lệnh từ sau dòng 35 này cho đến dòng 306, biến sRng của khối cũng xuất hiện trong khối các dòng lệnh I.  
Như vậy, có nghĩa là ta cần khai báo biến sRng dùng chung cho 2 macro cha & con.  
  
Để vậy, chúng ta khai báo biến sRng chỉ ngay sau dòng lệnh Option Explicit của module chứa cả hai macro. (Macro con để luôn trong macro sự kiện của trang tính ‘Phieu’)  
  
Tuy nhiên cũng cần mở ngoặt thêm chổ này để cùng rõ hơn:  
Vì trước khối lệnh I vùng Rng để tìm kiếm được gán cho các ô **Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))**  
Nhưng trước khối lệnh II, vùng Rng lại là **Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))**  
  
Nên sRng là ô tìm thấy tương ứng với 2 vùng trên sẽ khác cột nhau.   
Bỡi lẽ đó các giá trị cần gán vô form của trang tính ‘Phieu’ (vùng từ B7: H99) sẽ tương ứng khác nhau vế cột trong 2 đoạn mã giữa khối I & khối II  
  
Để cụ thể hơn, ta xét trường hợp mã hàng cần điền vô cột ‘C’ của trang tính ‘Phieu’  
  
Trường hợp ứng với khối dòng lệnh I, ta cần tìm tên các nhà cung cấp có tại cột ‘B’ của trang tính ‘PhatSinh’, thì mã hàng là cột liền kề bên fải so với ô được tìm thấy trên ‘B’;  
Nhưng trường hợp II thì ta tìm ngày nhập (Ở cột ‘A’ trên trang ‘PhatSinh’), lúc đó mã hàng là ô cách ô sRng tím thấy 1 cột.  
  
Qua đó chúng ta cũng sẽ thấy các trị cần gán vô form sẽ khác cột với nhau 1 cách tương ứng khi macro thực hiện các câu lệnh trong khối I & khối II.  
  
Qua thí dụ điển hình này, chúng ta sẽ rút ra kết luận, là cần cung cấp cho macro con trong trường hợp II các thông số về độ lệnh cột giữa lần chép I & lần chép II;   
Nhưng việc cung cấp này sẽ được mặc định 1 trị từ lệnh gọi (dòng lệnh số 2) bằng từ khóa Optional trong fần đầu của chương trình con. (Còn gọi macro con từ dòng lệnh 3 của macro cha sẽ fải cung cấp tham số đầy đủ).  
  
Mệnh đề **Optional Add1 As Byte** nói lên rằng: Tham số Add1 có thể được truyền, thì sẽ là trị số được truyền, nếu không được truyền thì tham số đó bằng 0 một cách mặc định  
  
Đó là chúng ta mới nói đến tham số có tên Add1 (Xem trong macro GPE\_Copy); Còn 1 tham số được truyền từ cha sang con nữa, đó là Offs. ( Hơn nữa, nếu không được truyền, tham số này mặc định là -1)  
  
Ta sẽ xét kỹ thêm về nó:  
  
Để vậy, chúng ta xét từ xuất fát điểm của ta là đi tìm ngày nhập hàng cụ thể nào đó của 1 NCC (nhà cung cấp) cụ thể nào đó. Mà ta biết rằng, ta có nhiều NCC cung ứng hàng nhiều lần trong tháng, thậm chí cung ứng nhiều lần trong ngày nữa & trong 1 ngày có thể có nhiều NCC nhập hàng. Bên trang CSDL dữ liệu về NCC được lưu ở cột ‘B’ còn ngày nhập hàng ghi ở cột ‘A’  
  
Khi áp dụng fương thức tìm kiếm 1 ngày nào đó, thì macro sẽ lọc ra DS (danh sách) NCC hôm đó; Nhiệm vụ tiếp theo là trong DS này, ta fải lựa ra 1 khách hàng ta cần mà thôi. Tất cả điều này liên quan đến khối lệnh tìm theo ngày;  
  
Còn khối lệnh tìm theo khách hàng cung ứng hàng trong tháng thì ngược lại chút đĩnh: Ta sẽ lọc ra từ DS các ngày trong tháng mà khách hàng này cung cấp hàng để chọn ra ngày cụ thể theo yêu cầu tại ô [H4]  
  
Nói nôm na là trường hợp đầu tứ cột ‘A’ ta tìm sang ‘B’ & ngược lại, trường hợp sau ta fải từ DS của cột ‘B’ sưu truy sang cột ‘A’  
  
Việc này sẽ fải nhờ đền biến Offs (để nó offset(,-1) hay Offset(,1) giúp ta trong trường hợp cụ thể gọi macro con từ dòng lệnh 2 hay từ dòng lệnh 3 từ macro cha)  
  
Nếu các bạn sưu tra thêm vế tham số Offs này, các bạn quan tâm đến nó đang trong vòng lặp Do . . . Loop mà thôi. (Còn dưới vòng lặp đó chẳng qua là tận dùng để làm việc khác mà khỏi khai báo thêm biến.)  
  
  
Kết quả của việc làm của chúng ta qua 3 bài là như thế này:  
  
**Macro cha:**

PHP Code:

Option Explicit  
 Dim Sh As Worksheet, Rng As Range, sRng As Range  
   
Private Sub Worksheet\_Change(ByVal Target As Range)  
 Dim NCC As Boolean, Ngay As Boolean  
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'  
   
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'  
 If Not Intersect(Target, [C4]) Is Nothing Then  'Nha CC'  
   [B7].Resize(14, 7).ClearContents  
   If [C4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))  
   If [H4].Value = "" Then Ngay = True  
   Set sRng = Rng.Find([C4].Value, , xlFormulas, xlWhole)  
2  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay  
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'  
   [B7].Resize(14, 7).ClearContents  
   If [H4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))  
   If [C4].Value = "" Then NCC = True  
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole)  
3  If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1  
 End If  
End Sub

**Còn đây là macro con (nằm luôn trong cửa sổ VBE sự kiện này luôn):**

Code:

**Sub GPE\_Copy(Cls As Range, All\_ As Boolean, Optional Offs As Integer = -1, Optional Add1 As Byte)**

Dim MyAdd As String

[B7].Resize(99).EntireRow.Hidden = False

MyAdd = sRng.Address: Application.ScreenUpdating = False

Do

With [b21].End(xlUp).Offset(1)

If All\_ Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

Else

If sRng.Offset(, Offs).Value = Cls.Value Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

End If

End If

End With

Set sRng = Rng.FindNext(sRng)

Loop While Not sRng Is Nothing And sRng.Address <> MyAdd

With [B99].End(xlUp)

If .Row < 13 Then Offs = 15 Else Offs = .Row + 2

End With

Range("B99:B" & Offs).EntireRow.Hidden = True

Set Sh = Nothing

**End Sub**

*(Chắc bài này sẽ khó hiểu & gây khó chịu với nhiều bạn; Xin lỗi vì sự diễn đạt kém của mình. Sẽ cố gắng theo dõi, bổ sung hay sửa chữa khi nhận được góp í từ bạn đọc - Rất cảm ơn!)*

**BÀI 21**:**BÀI TẬP ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TỰ NHAU** **TRONG MACRO.**

Chúng ta có macro sau (nguồn:= [http://www.giaiphapexcel.com/forum/s...ều-kiện.#6](http://www.giaiphapexcel.com/forum/showthread.php?48816-L%E1%BB%8Dc-d%E1%BB%AF-li%E1%BB%87u-tr%C3%AAn-4-%C4%91i%E1%BB%81u-ki%E1%BB%87n.#6) )

PHP Code:

Option Explicit  
Dim Sh As Worksheet:                        Dim eRw As Long  
Private Sub Worksheet\_Change(ByVal Target As Range)  
 Dim WF, Cls As Range, Rng As Range  
 Dim lRs As Long, Jj As Byte  
   
 Set WF = Application.WorksheetFunction  
 Set Rng = Sh.[B6].Resize(eRw, 10)  
 [b1:B99].EntireRow.Hidden = False  
1 'Xu Lí Vói Tháng Nam:'  
 If Not Intersect(Target, [e3]) Is Nothing Then  
    [b20].CurrentRegion.Offset(3, 1).ClearContents  
11 'Xu Lý Vói Tùng Tháng Cu The Trong Nam:'  
    If Target.Value <> "All" Then  
        With [B99].End(xlUp).Offset(1)  
            .Value = "'" & Right("0" & [e3].Value, 2) & "/" & [G3].Value  
5            .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[AA1].Resize(2, 2))  
            .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[AA1].Resize(2, 2))  
            .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[AA1].Resize(2, 2))  
        End With  
12 'Xu Lí Vói Toàn Bo Các Tháng Trong Nam'  
    Else  
        Sh.[ac1].Value = "Th"  
        For Jj = 1 To 12  
            Sh.[ac2].Value = Jj  
            With [B99].End(xlUp).Offset(1)  
                .Value = "'" & Right("0" & Sh.[ac2].Value, 2) & "/" & [G3].Value  
6                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[Ab1].Resize(2, 2))  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[Ab1].Resize(2, 2))  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[Ab1].Resize(2, 2))  
            End With  
        Next Jj  
    End If  
    Range([b11].End(xlDown).Offset(2), [B98]).EntireRow.Hidden = True  
2 'Xu Lý Vói Ten Dói Tác:'  
 ElseIf Not Intersect(Target, [e4]) Is Nothing Then  
    [b20].CurrentRegion.Offset(3, 1).ClearContents  
    Sh.[ac1].Value = Sh.[d6].Value  
21 'Xu Lý Voi Tùng Dói Tác:'  
    If Target.Value <> "All" Then  
        Sh.[ac2].Value = [G4].Value  
        If [e3].Value <> "All" Then  
            With [B99].End(xlUp).Offset(1)  
                .Value = "'" & Right("0" & [e3].Value, 2) & "/" & [G3].Value  
                .Offset(, 2).Value = [G4].Value  
                .Offset(, 3).Value = [e4].Value  
7                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[AA1].Resize(2, 3))  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[AA1].Resize(2, 3))  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[AA1].Resize(2, 3))  
            End With  
        Else  
            Sh.[ad1].Value = "Th"  
            For Jj = 1 To 12  
                Sh.[ad2].Value = Jj  
                With [B99].End(xlUp).Offset(1)  
                    .Value = "'" & Right("0" & Jj, 2) & "/" & [G3].Value  
                    .Offset(, 2).Value = [G4].Value  
                    .Offset(, 3).Value = [e4].Value  
8                    .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[Ab1].Resize(2, 3))  
                    .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[Ab1].Resize(2, 3))  
                    .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[Ab1].Resize(2, 3))  
                End With  
            Next Jj  
        End If  
22 'Xu Lý Vói Tát Ca Các Dói Tác:'  
    Else  
        Dim Nam As String, Crit As Range  
   
        If [e3].Value <> "All" Then  
            Set Crit = Sh.[AA1].Resize(2, 3)  
            Nam = Right([C3].Value, 5) & " " & [e3].Value & "/" & [G3].Value  
        Else  
            Set Crit = Sh.[Ab1].Resize(2, 2)  
            Nam = Right([b11].Value, 3) & " " & [G3].Value  
        End If  
        For Each Cls In Sheets("DKien").Range("MaDT").SpecialCells(xlCellTypeConstants, 2).Offset(1)  
            If Cls.Value = "All" Then Exit For  
            Sh.[ac2].Value = Cls.Value  
            With [B99].End(xlUp).Offset(1)  
                .Value = Nam  
                .Offset(, 2).Value = Cls.Value  
                .Offset(, 3).Value = Sheets("DKien").Range("MaDT"). \_  
                    Find(Cls.Value, , xlFormulas, xlWhole).Offset(, -1).Value  
9                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Crit)  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Crit)  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Crit)  
            End With  
        Next Cls  
    End If  
    Range([b11].End(xlDown).Offset(2), [B98]).EntireRow.Hidden = True  
 End If  
End Sub

Quan sát macro ta thấy 3 dòng lệnh kế tiếp của các dòng lệnh được đánh số từ 5 đến 9 có cấu trúc na ná nhau;  
  
Nhiệm vụ đề ra là ta thử gộp chúng vô 1 macro con như cách thúc mà các bài 1 đến 3 về đặt thừa số chung mà ta đã xét qua.  
(Tất nhiên bạn có thể tham khảo cách làm của bài tiếp sau trong topic đó) 