**Bài 10. Array**

|  |
| --- |
| **0. Khái quát chung**  Mảng (array) là một tập hợp của nhiều phần tử được gọi thông qua một tên.  Ví dụ: Arr  Ta xác định phần tử trong mảng thông qua địa chỉ của phần tử đó trong mảng.  Ví dụ: Arr(2): Tức là gọi tới phần tử có địa chỉ thứ tự là 2 trong mảng Arr.  Phân loại:  Theo kích thước của mảng, chia thành: mảng 1 chiều và mảng nhiều chiều.  Ví dụ:  Mảng 1 chiều: Arr(5)  Mảng 2 chiều: Arr2(4, 9)  Mảng 3 chiều: Arr3(5,12,30)  Theo tính chất về khả năng thay đổi (khai báo lại) kích thước của mảng trong quá trình sử dụng, chia thành: mảng tĩnh, mảng động.  Khai báo mảng:  Public | Private | Dim | Static Arr([Lower To] Upper) [As type of variables]  - Public | Private | Dim | Static: Là các từ khóa để khai báo biến mảng.  (xem chi tiết ở [Bài 4. Biến trong VBA](https://sites.google.com/site/exl2lab/vba-in-excel/Bai-4-bien-trong-vba))  - Arr: Tên của biến mảng cần khai báo, tuân theo quy tắc đặt tên một biến.  - Lower: Cận dưới một chiều của mảng (\*).  - Upper: Cận trên một chiều của mảng  - To: Là từ khóa  - As type of variables: Kiểu dữ liệu của các phần từ trong mảng Arr.  Nếu mảng gồm nhiều chiều thì các chiều được ngăn cách bởi dấu phẩy.  (\*) Chú thích: Cận dưới một chiều của mảng mặc định Lower=0, tức là mặc định trong VBA Option Base 0.  Ví dụ:           Dim Arr1(5): Tương ứng với Dim Arr1(0 To 5), mảng 1 chiều có 6 phần tử.           Dim Arr2(6, 9): Tương ứng với Dim Arr(0 To 6, 0 To 9), mảng 2 chiều, chiều thứ nhất (0-6), chiều thứ hai (0-9).  Để thay đổi cận dưới mặc định này ta khai báo ở trên cùng của module cần thay đổi:           Option Base 1  Khi đó: Dim Arr1(5) sẽ tương ứng Dim Arr1(1 To 5).  Kích thước của mảng:  Dùng 2 hàm LBound() và UBound() để xác định cận dưới và cận trên của mảng.  LBound(ArrayName, Dimension)   UBound(ArrayName, Dimension  ArrayName: Tên mảng.  Dimension: Chiều của mảng cần xác định.  Kích thước của một chiều thứ i của mảng Arr = UBound(Arr,i) – Lbound(Arr,i) + 1. Kích thước lớn nhất mỗi chiều của mảng là 2^31 – 1. Tuy nhiên, mảng đó có hoạt động được hay không còn phụ thuộc vào bộ nhớ của máy tính, nếu mảng có kích thước vượt quá bộ nhớ khả dụng của máy tính sẽ gặp lỗi “Out of memory”. Tham khảo thêm [Array Size](https://msdn.microsoft.com/en-us/library/b388cb5s(v=vs.90).aspx). Ví dụ:  Sub L\_UBound()     Dim Arr(1 To 10, 1 To 4)     MsgBox LBound(Arr, 1) & " - " & UBound(Arr, 1)  'Result: 1 - 10     MsgBox LBound(Arr, 2) & " - " & UBound(Arr, 2) 'Result: 1- 4 End Sub  Ứng dụng của mảng trong viết code: Tăng tốc cho code.  Theo tính chất khi sử dụng, ta xét mảng tĩnh và mảng động.  **1. Mảng tĩnh**  Một mảng tĩnh là mảng có số chiều, kích thước của chiều được khai báo cụ thể ngay từ đầu.  Dim StaticArr (1 đến 20) As Long  Không thể thay đổi kích thước hoặc kiểu dữ liệu của mảng tĩnh (Redim / Redim Preserve).  Khi thực hiện xóa một mảng tĩnh (Erase StaticArr), không có bộ nhớ nào được giải phóng. Khi đó, việc xóa đó đơn giản là đặt lại tất cả các phần tử của mảng về giá trị mặc định của chúng (0, vbNullString, Empty hoặc Nothing, tùy thuộc vào kiểu dữ liệu của mảng đó đã được khai báo).  Ví dụ:  Dim Arr(1 To 8) 'Mảng 1 chiều có 8 phần từ (1-8)   Dim Arr2(1 to 9, 1 To 1) 'Mảng 2 chiều, chiều 1 có 9 phần tử (1-9), chiều 2 có 1 phần từ (1-1)  Kích thước khai báo phải là một hằng số cụ thể, không thể khai báo thông qua một biến.  Đúng:  Dim A(1 to 3)  Dim A1( 1 to 4, 1 to 9)  Không đúng:  Sub KhaiBaoLoi()     Dim n As Long     n = 5     Dim Arr(1 To n) 'Error: Constant expression required End Sub  **2. Mảng động**  Khi chưa xác định được số chiều và kích thước chiều của mảng cần khai báo, ta dùng mảng động. Mảng động cho phép khai báo số chiều, kích thước mỗi chiều của mảng sau.  Ví dụ:  Dim Arr1() As Long, Arr2() As String   'Ta mới chỉ khai báo biến mảng nhưng chưa biết số chiều, kích thước của chiều.'  Mảng động cho phép thay đổi kích thước của mảng, nhưng không thể thay đổi kiểu dữ liệu của mảng.  Khi thực hiện xóa (Erase) một mảng động, bộ nhớ được phân bổ cho mảng đó được giải phóng. Để sử dụng mảng động sau khi Erase cần thực hiện ReDim mảng động đó.  Khi đã xác định được số chiều và kích thước của chiều ta tiến hành khai báo lại cho mảng, với từ khóa Redim  Ví dụ:  Sub Arr\_Dynamic\_1()     Dim Arr1() As Long, Arr2() As Long, m As Long, n As Long, i As Long     m = 50     n = 60     ReDim Arr1(1 To m)          '1.1'     ReDim Arr2(1 To m, 1 To 1)  '1.2'     For i = 1 To m              'Vòng lap gán giá tri vào mang'         Arr1(i) = i         Arr2(i, 1) = i     Next i     ReDim Arr1(1 To n)          '2.1'     ReDim Arr2(1 To m, 1 To n)  '2.2'     MsgBox "Arr1(1) = " & Arr1(1) & vbNewLine & "Arr2(1,1) = " & Arr2(1, 1)     'Result: Arr1(1) = 0  |  Arr2(1,1) = 0' End Sub  Trong ví dụ trên, ban đầu ta khai báo 2 biến mảng Arr1(), Arr2() nhưng chưa khai báo số chiều và kích thước của chiều.  Tiếp đó, ta khai báo lại lần thứ nhất (1.1 và 1.2) 2 mảng đó với số chiều và kích thước cụ thể:  ReDim Arr1(1 To m): mảng có 1 chiều, có m -1 +1 = 50 phần tử.  ReDim Arr2(1 To m, 1 To 1): mảng 2 chiều, chiều 1 có kích thớc m -1 +1 = 50, chiều 2 có kích thớc là 1.  Tiếp theo, ta gán các giá trị cho các phần tử trong mảng (vòng lặp For Next).  Sau đó, ta khai báo lại lần thứ hai (2.1 và 2.2) 2 mảng đó với số chiều và kích thước cụ thể:  ReDim Arr1(1 To n): mảng có 1 chiều, có n -1 +1 = 60 phần tử.  ReDim Arr2(1 To m, 1 To n): mảng 2 chiều, chiều 1 có kích thớc m -1 +1 = 50, chiều 2 có kích thước n -1 +1 = 60.  Lưu ý ở đây:  - Sau khi Redim để khai báo thay đổi số chiều và kích thước của chiều thì các phần tử trong mảng trước đó đều bị xóa.  - Khi cần khai báo lại để thay đổi kích thước của mảng mà vẫn giữ các phần từ đã có thì ta dùng Redim Preserve. Tuy nhiên, cần chú ý rằng Redim Preserve chỉ áp dụng được để thay đổi kích thước chiều cuối cùng của mảng.  Ví dụ:  Sub Arr\_Dynamic\_2()     Dim Arr1() As Long, Arr2() As Long, m As Long, n As Long, i As Long     m = 50     n = 60     ReDim Arr1(1 To m)          '1.1'     ReDim Arr2(1 To m, 1 To 1)  '1.2'     For i = 1 To m              'Vòng lap gán giá tri vào mang'         Arr1(i) = i         Arr2(i, 1) = i     Next i     ReDim Preserve Arr1(1 To n)             '2.1'     ReDim Preserve Arr2(1 To m, 1 To n)     '2.2'     MsgBox "Arr1(1) = " & Arr1(1) & vbNewLine & "Arr2(1,1) = " & Arr2(1, 1)     'Result: Arr1(1) = 1  |  Arr2(1,1) = 1' End Sub  Để ý (2.2) ở ví dụ trên: ReDim Preserve Arr2(1 To m, 1 To n), chỉ có thể thay đổi kích thước chiều cuối (chiều thứ 2) của mảng Arr2.  **3. Chép trị từ range của bảng tính vào array và gán các phần tử của array xuống range của bảng tính**  (Range – Array – Range)  3.1. Range – Array:  Theo hình học giải tích ta có chia ra 1 chiều (đường thẳng), 2 chiều (mặt phẳng), 3 chiều (dạng khối), và n chiều.  Một range trên bảng tính Excel luôn luôn có 2 chiều (row và column), vậy range sẽ thuộc dạng mặt phẳng.  Khi chép giá trị của một range cho mảng (mảng động), VBA dùng một hàm copy. Hàm này mặc định cho kết quả là mảng 2 chiều, row của range tương ứng với chiều thứ nhất của mảng, column tương ứng với chiều thứ hai của mảng.  Dim Arr(), Rng As Range  'Set Rng = Range("A1:E20")' Arr = Rng.Value  Lưu ý:  - Trường hợp đặc biệt, khi range chỉ gồm 1 cell, VBA tự nhận biết và không dùng hàm copy mảng mà dùng hàm copy giá trị đơn, kết quả trả về là 1 giá trị (không phải mảng).  Do đó, phải bẫy lỗi trường hợp range chỉ gồm 1 cell trước khi copy vào mảng.  Dim Arr(), Rng As Range  'Set Rng = Range' If Rng.Count > 1 Then  Arr = Rng.Value  End If  Hoặc ta khai báo biến Arr ở dạng Variant, và kiểm tra kiểu dữ liệu của biến Arr sau khi được gán giá trị của range:  Dim Arr As Variant, Rng As Range 'Set Rng = Range("A1") Arr = Rng.Value If TypeName(Arr) = "Variant()" Then 'or: If IsArray(Arr) = True Then     'Your code.' End If  - Khi kết hợp dùng hàm Application.Transpose() áp dụng vào range chỉ có 1 cột thì kết quả trả về là mảng 1 chiều. Lưu ý: Hàm Transpose sẽ gặp lỗi khi có một phần tử trong mảng có nhiều hơn 255 ký tự.  Dim Arr(), Rng As Range Set Rng = Sheet1.Range("A2:A30") If Rng.Count > 1 Then     Arr = Application.Transpose(Rng.Value) End If  Ngoài ra, hàm Transopse chuyển mảng 1 chiều thành mảng 2 chiều: Arr2D=Application.Transpose(Arr1D)  - Cận dưới (LBound) của các chiều của mảng luôn bắt đầu từ 1, cho dù không khai báo Option Base 1.  - Không thể copy giá trị của một range vào mảng tĩnh theo cấu trúc Arr = Rng.Value, mà phải dùng vòng lặp để truyền từng giá trị của range vào mảng.  3.2. Array – Range:  Khi cần gán mảng xuống range ta dùng cấu trúc:  Rng.value = Arr  Trong đó, range có số dòng/ cột tương ứng (bằng hoặc nhỏ hơn) so với kích thước 2 chiều của mảng Arr.  Lưu ý:  - Khi gán mảng 1 chiều xuống range thì range đó chỉ gồm 1 dòng.  Sub ImportArray\_1D()     Dim Arr(), Rng As Range     Set Rng = Sheet1.Range("A2:A30")     If Rng.Count > 1 Then         Arr = Application.Transpose(Rng.Value) 'Arr là mang 1 chiêu'     End If     'Gán mang Arr xuông bang tinh'     Sheet1.Range("B10").Resize(1, UBound(Arr, 1)).Value = Arr End Sub  - Khi range chỉ định để gán giá trị của 1 mảng xuống mà nó có số dòng/ cột lớn hơn so với kích thước chiều của mảng thì những cells nằm ngoài sẽ trả về #N/A.  Sub ImportArray\_2D()     Dim Arr()     Arr = Sheet1.Range("A2:B4").Value     'Gán mang Arr xuông bang tinh, chú ý xem kêt qua #N/A:'     Sheet1.Range("A15").Resize(UBound(Arr, 1) + 2, UBound(Arr, 2) + 1).Value = Arr End Sub  **4. Ví dụ**  Xét ví dụ nhỏ: Điền số thứ tự dạng số nguyên (1-n) vào vùng A2:A1000000.  Function tạo trả về một mảng 2 chiều, chiều thứ nhất có số phần từ bằng số dòng của range, chiều thứ hai có kích thước bằng 1 phần tử.  Function ArrSeries(ByVal Rng As Range) As Variant     Dim aTmp(), i As Long     ReDim aTmp(1 To Rng.Rows.Count, 1 To 1)     For i = 1 To UBound(aTmp, 1)         aTmp(i, 1) = i     Next i     ArrSeries = aTmp End Function  Sub gọi function ArrSeries() rồi gán giá trị mảng trả về từ function ArrSeries() xuống range:  Sub SetSeries\_Array()     Dim Rng As Range, T As Double     T = Timer     Set Rng = Sheet1.Range("A1:A100000")     Rng.Value = ArrSeries(Rng)     MsgBox Round(Timer - T, 2) & " giây"    'T=0.12 giây' End Sub  Sub thực hiện gán số thứ xuống range theo phương thức gán xuống từng cell của range:  Sub SetSeries\_Range()     Application.ScreenUpdating = False     Application.Calculation = xlCalculationManual     Dim i As Long, T As Double     T = Timer     For i = 1 To 100000         Sheet1.Range("A1").Offset(i - 1, 0).Value = i     Next i     Application.ScreenUpdating = True     Application.Calculation = xlCalculationAutomatic     MsgBox Round(Timer - T, 2) & " giây"    'T=2.75 giây' End Sub |