

BÀI THỰC HÀNH 7

Ôn Tập



1. Hãy **ĐOC CÂN THÂN TẤT CẢ NỘI DUNG** trong bài thực hành trước khi làm bài.
2. Sinh viên **TỰ GỖ CODE, KHÔNG COPY** từ bài có sẵn, để nắm được việc lập trình.

1 Chuẩn đầu ra

Củng cố lại các kiến thức về lập trình với ngôn ngữ Visual Basic đã được học.

2 Chuẩn bị

Đọc lại các kiến thức lập trình với ngôn ngữ Visual Basic đã được học.

Tài liệu tham khảo gợi ý: Giáo trình Visual Basic 6.0, Nguyễn Đăng Quang, Đại học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.

3 Phương tiện

- Máy tính có cài đặt Visual Studio 6.0.

4 Thời lượng: 4 tiết

5 Nội dung thực hành

5.1 Vòng lặp	2
5.2 Mảng.....	3
5.3 Xử lý chuỗi.....	4
5.4 Tham trị và tham chiếu (ByVal và ByRef)	5
5.5 ListBox và ComboBox.....	6

5.1 VÒNG LẶP

5.1.1 Viết chương trình tính giá trị các biểu thức sau đây:

$$S = 1/2 + 2/3 + \dots + n/(n+1)$$

$$S = -1 + 2 - 3 + 4 - \dots + (-1)^n * n$$

$$S = 1! + 2! + \dots + n!$$

với n được nhập bởi người dùng.

Hướng dẫn: xem lại bài 6.4 Tính giá trị biểu thức với sai số cho trước, trong BÀI THỰC HÀNH 2.

5.1.2 Viết chương trình tìm ước chung lớn nhất, bội chung nhỏ nhất của 2 số a, b được nhập bởi người dùng. tối giản phân số.

Hướng dẫn:

- Tìm Ước chung lớn nhất (UCLN): dùng **Thuật toán Euclid**:

Để tìm UCLN của 2 số nguyên dương a, b ($a > b$), ta tìm UCLN của b và số dư của a chia b:

$$UCLN(a, b) = UCLN(b, a \bmod b)$$

Hay nói cách khác: ta tìm số dư q của a/b, sau đó lại tìm số dư của b/q. Lặp lại việc tìm số dư này cho đến khi số dư bằng 0, khi đó số dư khác không cuối cùng chính là UCLN của a và b.

Ví dụ: tìm UCLN (128, 48):

$$UCLN(128, 48) = UCLN(48, 32) = UCLN(32, 16) = UCLN(16, 0)$$

Hay: $128 \rightarrow 48 \rightarrow 32 \rightarrow 16 \rightarrow 0$ (tìm liên tiếp các số dư)

$$\rightarrow UCLN(128, 48) = 16$$

- Tìm Bội chung nhỏ nhất (BCNN):

$$BCNN \text{ của } 2 \text{ số nguyên dương } a, b = a*b/UCLN(a, b)$$

5.1.3 Viết chương trình thực hiện tối giản phân số a/b, với a, b là số nguyên dương được nhập bởi người dùng.

Hướng dẫn: để tối giản phân số a/b ta tìm Ước chung lớn nhất của a và b, sau đó chia cả a và b cho UCLN ta sẽ có phân số tối giản.

$$\text{Ví dụ: } 128/48 = (128/16) / (48/16) = 8/3 \text{ (chú ý: } UCLN(128, 48) = 16)$$

5.2 MẢNG

5.2.1 Viết chương trình in ra màn hình số lần xuất hiện của các phần tử trong mảng, với mảng có số phần tử tùy ý được nhập bởi người dùng. Dùng InputBox để nhập các phần tử cho mảng.

Ví dụ: Mảng: 3 4 3 7 5 7 7

- Số 3: xuất hiện 2 lần
- Số 4: xuất hiện 1 lần
- Số 5: xuất hiện 1 lần
- Số 7: xuất hiện 3 lần

Hướng dẫn: dùng **mảng động** (xem lại BÀI THỰC HÀNH 4). Có thể thực hiện bằng cách sắp xếp mảng, rồi sau đó đếm số lần xuất hiện.

5.2.2 Viết chương trình kiểm tra một mảng số nguyên có phải là cấp số cộng hay không. Nếu là cấp số cộng thì tìm công sai của cấp số cộng này.

Ví dụ:

- Mảng: 3 4 3 7 5 7 7 → Không là cấp số cộng.
- Mảng: 5 7 9 11 13 15 → Là cấp số cộng, với công sai là 2.

5.2.3 Viết chương trình cho phép người dùng nhập vào một mảng với số lượng phần tử tùy ý, rồi sắp xếp mảng tăng dần. Sau đó chèn thêm một phần tử vào mảng sao cho mảng vẫn giữ thứ tự tăng dần.

Ví dụ: Mảng: 3 7 6 5 3 2 1

→ Sắp xếp tăng: 1 2 3 3 5 6 7

→ Chèn thêm phần tử **4**: 1 2 3 3 **4** 5 6 7

Hướng dẫn: dùng **mảng động** (xem lại BÀI THỰC HÀNH 4). Có thể thực hiện chèn thêm phần tử bằng cách sau:

- Trước tiên ReDim lại mảng để tăng số phần tử của mảng thêm 1. Cần dùng **ReDim Preserve** để giữ lại các phần tử đã có trong mảng (xem BÀI THỰC HÀNH 4).

Mảng ban đầu:

1	2	3	3	5	6	7
---	---	---	---	---	---	---

ReDim:

1	2	3	3	5	6	7	
---	---	---	---	---	---	---	--

- Sau đó duyệt mảng **từ cuối mảng** và dời các phần tử **lớn hơn phần tử cần chèn** lùi ra sau 1 vị trí.

1	2	3	3	5	6	<u>7</u>	
---	---	---	---	---	---	----------	--

1	2	3	3	5	<u>6</u>		7
---	---	---	---	---	----------	--	---

1	2	3	3	<u>5</u>		6	7
---	---	---	---	----------	--	---	---

1	2	3	3		5	6	7
---	---	---	---	--	---	---	---

(Phần tử cần chèn: 4)

- Cuối cùng, đưa phần tử cần chèn vào:

1	2	3	3	<u>4</u>	5	6	7
---	---	---	---	----------	---	---	---

5.3 XỬ LÝ CHUỖI

5.3.1 Viết chương trình đếm số từ có trong chuỗi (các từ được cách nhau bởi ít nhất một khoảng trắng). Chỉ được dùng hàm Mid và Len.

Ví dụ: Chuỗi: “DH Su Pham Ky Thuat TP.HCM” có 6 từ.

Hướng dẫn: có thể thực hiện như sau: trước tiên xóa các khoảng trắng thừa trong chuỗi (xem bài 6.6 Xóa khoảng trắng thừa trong chuỗi, trong BÀI THỰC HÀNH 3), sau đó đếm số khoảng trắng. Khi đó, số từ = số khoảng trắng + 1.

5.3.2 Viết chương trình thực hiện đảo thứ tự các từ trong chuỗi. Chỉ được dùng hàm Mid và Len.

Ví dụ: Chuỗi: “Lỗi người không thấy xét thân ta” → đảo thành: “Ta thân xét thấy không người lỗi”

Hướng dẫn: có thể thực hiện như sau: trước tiên xóa các khoảng trắng thừa trong chuỗi (xem bài 6.6 Xóa khoảng trắng thừa trong chuỗi, trong BÀI THỰC HÀNH 3), sau đó dùng vòng lặp duyệt chuỗi từ cuối chuỗi, khi nào gặp một khoảng trắng thì đó là một từ, ta tiến hành cắt từ này (dùng hàm Mid) và đưa vào chuỗi mới.

5.3.3 Viết chương trình tìm và thay thế một từ trong chuỗi bằng từ khác. Chỉ được dùng hàm Mid và Len.

Ví dụ: Chuỗi: “DH SP Ky Thuat” → thay thế “SP” bằng “Su Pham” → chuỗi sau khi thay thế: “DH Su Pham Ky Thuat”.

Hướng dẫn:

- Dùng vòng lặp duyệt từ đầu chuỗi cho đến khi tìm thấy chuỗi cần tìm.

s = "DH **SP** Ky Thuat"
 ↑
 i=4

- Sau đó, cắt chuỗi ban đầu thành 2 phần (bỏ đi phần bị thay thế) và đưa chuỗi thay thế vào (xem lại BÀI THỰC HÀNH 3 về xử lý chuỗi).

```
snew = "DH " & "Su Pham" & " Ky Thuat"
(snew = Mid(s, 1, i-1) & "Su Pham" & Mid(s, i+Len("SP")))
→ snew = "DH Su Pham Ky Thuat"
```

5.4 THAM TRỊ VÀ THAM CHIẾU (BYVAL VÀ BYREF)

Hướng dẫn: Xem lại BÀI THỰC HÀNH 3 về Tham trị và Tham chiếu.

Lưu ý: Sinh viên **tự chạy tay** chương trình để tìm kết quả (**không gõ code vào máy tính**). Sau đó có thể gõ code vào máy để kiểm tra kết quả.

5.4.1 Cho biết giá trị của 2 biến a và b khi chạy thủ tục cmdTest1_Click trong đoạn chương trình sau:

```
Private Function Ham1 (ByRef a As Long, ByVal b As Long) As Long
    a = a + b
    a = a * a
    Ham1 = a
End Function
Private Sub ThuTuc2 (ByVal a As Long, ByRef b As Long)
    b = a + b
    a = Sqr(a)
End Sub

Private Sub cmdTest1_Click()
    Dim a As Long, b As Long
    a = 2
    b = 3

    a = Ham1(b, a)
    ThuTuc2 a, b
    MsgBox "a = " & a & ", b = " & b
End Sub
```

5.4.2 Cho biết giá trị của 2 biến a và b khi chạy thủ tục cmdTest2_Click trong đoạn chương trình sau:

```
Private Function Ham1 (ByVal a As Long, ByRef b As Long) As Long
    b = a + b
    a = a * b
    Ham1 = a
End Function
Private Sub ThuTuc2 (ByVal a As Long, ByRef b As Long)
    b = a - b
    a = Sqr(a)
End Sub

Private Sub cmdTest2_Click()
    Dim a As Long, b As Long
    a = 2
    b = 3

    a = Ham1(a, b)
    ThuTuc2 b, a
    MsgBox "a = " & a & ", b = " & b
End Sub
```

5.5 LISTBOX VÀ COMBOBOX

5.5.1 Viết chương trình quản lý bán hàng dùng các ListBox và ComboBox như sau:

Yêu cầu:

- ListBox *lstTenSanPham* hiển thị tên các sản phẩm. Mỗi sản phẩm sẽ có đơn giá được lưu trong **ItemData** của nó.
- Khi người dùng chọn một sản phẩm trong ListBox *lstTenSanPham* và chọn số lượng trong ComboBox *cboSoLuong* và nhấn nút “>”, sản phẩm sẽ được ghi vào ListBox *lstDatHang*, đồng thời giá tiền của nó (**đơn giá x số lượng**) sẽ được ghi vào ListBox *lstThanhTien*.
- Khi người dùng chọn một mục trong ListBox *lstDatHang* và nhấn nút “<” thì mục này sẽ bị xóa khỏi ListBox *lstDatHang* và mục tương ứng trong ListBox *lstThanhTien* cũng bị xóa.
- Khi người dùng cuộn ListBox *lstDatHang* hoặc ListBox *lstThanhTien* thì ListBox kia sẽ tự động cuộn theo.
- Khi người dùng click chọn một mục trong ListBox *lstDatHang* thì mục tương ứng trong ListBox *lstThanhTien* cũng được chọn.

Hướng dẫn: xem lại BÀI THỰC HÀNH 4 và BÀI THỰC HÀNH 5 về ListBox và ComboBox.

_____HẾT_____