BÀI THỰC HÀNH 3

Hàm và Thủ Tục, Biến Toàn Cục và Cục Bộ, Tham Trị và Tham Chiếu, Chuỗi, Mảng

1. Hãy <u>ĐỌC CẨN THẬN TẤT CẢ NỘI DUNG</u> trong bài thực hành trước khi làm bài. 2. Sinh viên <u>TỰ GÕ CODE, KHÔNG COPY</u> từ bài có sẵn, để nắm được việc lập trình.

1 Chuẩn đầu ra

Sau bài này, người học có thể:

- Phân biệt được các phạm vi của biến: toàn cục, cục bộ (Public, Private).
- Phân biệt và sử dụng được Hàm (Function) và Thủ tục (Sub).
- Chỉ ra được kết quả khác nhau khi truyền tham số bằng Giá trị và Địa chỉ (ByVal và ByRef).
- Sử dụng được cú pháp bẫy lỗi On Error.
- Sử dụng được các kiểu dữ liệu có cấu trúc: Chuỗi, Mảng.

2 Chuẩn bị

Đọc lại về Hàm và Thủ tục, Phạm vi của biến, Cách truyền tham số, Các kiểu dữ liệu có cấu trúc (Chuỗi, Mảng).

Tài liệu tham khảo gọi ý: Chương 3, Chương 4, Chương 5, 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 Kiến thức bổ trợ

- Các hàm sử dụng:
 - Left(*s*,*n*): trả về *n* ký tự đầu tiên bên trái chuỗi *s*. Ví dụ: Left("Visual Basic",6)→"Visual"
 - Right(*s*,*n*): trả về *n* ký tự đầu tiên bên phải chuỗi *s*.
 Ví dụ: Right("Visual Basic",5)→"Basic"
- Giải thuật sắp xếp Chọn (Selection sort): Cho một mảng các phần tử:
 - 1. Chọn phần tử đầu tiên.
 - So sánh với các phần tử sau nó trong mảng, nếu không thỏa điều kiện sắp xếp thì đổi chỗ. Ví dụ: phần tử đầu tiên: 4, một phần tử sau nó: 3, yêu cầu sắp tăng dần → phải đổi chỗ 3 và 4.
 - 3. Chọn phần tử kế tiếp, lặp lại bước 2, cho đến khi hết mảng.

- Tham trị và tham chiếu: Có 2 kiểu truyền tham số cho hàm/thủ tục: tham trị và tham chiếu:
 - Tham trị (ByVal): Giá trị truyền vào được copy thành một biến khác để đưa vào hàm/thủ tục → Giá trị của biến dùng làm tham số cho hàm/thủ tục không bị thay đổi.
 - Tham chiếu (ByRef): Giá trị truyền vào không được copy thành 1 biến khác. Hàm/Thủ tục làm việc trực tiếp trên biến làm tham số → Giá trị của biến làm tham số có thể bị thay đổi bởi hàm/thủ tục. Đây là kiểu truyền tham số mặc định của VB.
- Các hệ đếm:
 - Hệ thập phân: hay hệ đếm cơ số 10 (decimal), dùng 10 chữ số (0, 1, 2, 3, 4, 5, 6, 7, 8 và 9) để biểu đạt giá trị. Ví dụ: 34; 5.11
 - Hệ nhị phân: hay hệ đếm cơ số 2 (binary), dùng 2 chữ số (0 và 1) để biểu đạt giá trị. Ví dụ: 101011; 10.011
 - *Hệ thập lục phân*: hay hệ đếm cơ số 16 (hexadecimal), *dùng 16 chữ số* (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) để biểu đạt giá trị. Ví dụ: A25; E34F; 4AC0.A3
- Cách chuyển đổi giữa các hệ đếm: có 3 nguyên tắc:

<u>Nguyên tắc 1</u>: Để đổi từ các hệ khác sang thập phân: ta nhân từng trọng số với lũy thừa bậc n của cơ số, với n là chỉ số vị trí (vị trí hàng đơn vị là 0).

Ví dụ:

• Đổi số nhị phân (**co** số = 2) 1101.011₂ sang hệ thập phân:

	Trọng số	1	1	0	1	0	1	1	
	Chỉ số vị trí	3	2	1	0	-1	-2	-3	
→ 1101.011 ₂ =	$=1^{*2^{3}}+1^{*2^{2}}+0$	*2 ¹	+ 1	*2 ⁰	+ 0	*2 ⁻¹	+ 1*	2 ⁻² +	- 1* 2⁻³
=	= 13.375 ₁₀								

• Đổi số thập lục phân (**cơ số = 16**) $4AB.3_{16}$ sang hệ thập phân:

	Trọng số	4	А	В	3
	Chỉ số vị trí	2	1	0	-1
→ 4AB.3 ₁₆ = $4*16^2$ +	10* 16 ¹ + 11* 1 6	5 ⁰ +	3*1	6 ¹	
= 1195.18	75 ₁₀				

<u>Nguyên tắc 2</u>: Để đổi từ thập phân sang các hệ khác: ta chia số cần đổi cho cơ số hệ đích (2; 16), ghi lại số dư, sau đó lấy phần nguyên của thương chia tiếp cho cơ số hệ đích. Lặp lại đến khi phần nguyên của thương = 0. Kết quả là các số dư ghi ngược từ dưới lên. Nguyên tắc này dùng cho số nguyên.

Đối với số có phần thập phân, ta đổi phần nguyên như trên. Còn *phần thập nhân* thì ta *nhân* với cơ số hệ đích, ghi lại phần nguyên của tích, sau đó lấy phần thập nhân của tích tiếp tục nhân với cơ số đích cho đến khi phần thập phân này = 0. Kết quả là các phần nguyên ghi từ trên xuống.

Ví dụ: Đổi số thập phân 13.375_{10} sang hệ nhị phân (cơ số đích = 2):

 \circ Đổi phần nguyên 13₁₀: *chia 2 lấy du*



→ $13_{10} = 1101_2$

• Đổi phần thập phân 0.375₁₀: nhân 2 lấy phần nguyên

 $\begin{array}{cccc}
0.375 * 2 &= & 0.75 \\
0.75 * 2 &= & 1.5 \\
0.5 * 2 &= & 1.0
\end{array}$

→ $0.375_{10} = 0.011_2$

Vậy $13.375_{10} = 1101.011_2$

<u>Nguyên tắc 3</u>: Chuyển đổi giữa nhị phân và thập lục phân: mỗi chữ số thập lục phân $\Leftrightarrow 4$ chữ số nhị phân.

Bảng sau liệt kê các giá trị tương ứng giữa 2 hệ thập lục phân và nhị phân:

Thập lục phân	Nhị phân
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
А	1010
В	1011
С	1100
D	1101
E	1110
F	1111

Ví dụ:

 \circ Đổi số nhị phân 1100111.011₂ sang thập lục phân:

 $1100111.011_2 = 0110\ 01111\ .\ 0110_2 = 67.6_{16}$

 \circ Đổi số thập lục phân 2C5.7E₁₆ sang nhị phân:

 $2C5.7E_{16} = 0010\ 1100\ 0101\ .\ 0111\ 1110_2 = 1011000101.0111111_2$

5 Thời lượng: 4 tiết

6 Nội dung thực hành

Bài thực hành có hướng dẫn chi tiết:

6.1	Chuyển đổi giữa thập phân và thập lục phân	4
6.2	Tính tổ hợp	9
6.3	Tham trị và tham chiếu1	1
6.4	Sắp xếp mảng tăng dần1	3
6.5	Bẫy lỗi bằng On Error1	5
6.6	Xóa khoảng trắng thừa trong chuỗi1	8
Bài thự	rc hành tự thực hiện:	
7.1 và n	Viết chương trình chuyển đổi số giữa các hệ đếm: nhị phân <-> thập phâr hị phân <-> thập lục phân1	ı, 9
7.2	Viết chương trình đếm số từ trong chuỗi1	9
7.3 khoả	Viết chương trình sắp xếp mảng tăng dần, với mảng là một chuỗi các số ngăn cách bằn ảng trắng*1	g 9

6.1 CHUYỂN ĐỔI GIỮA THẬP PHÂN VÀ THẬP LỤC PHÂN

6.1.1 Mô tả

Dữ liệu vào: một số thập phân hoặc thập lục phân (nguyên dương).

Dữ liệu ra: chuyển đổi thành số thập lục phân hoặc thập phân.

Điều khiển: 2 nút "Chuyen sang THAP LUC PHAN" và "Chuyen sang THAP PHAN": đổi số sang hệ tương ứng.

6.1.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai12_ChuyenDoiThapPhan_ThapLucPhan (tự tạo trước).
- 1. Tạo form như hình:

		CommandButton: cmdThapPhanSangT cmdThapLucPhanSa	ThapLucPhan IngThapPhan	
	🖏 Chuyen doi Th	ap phan <-> Thap luc pł	ian 📃	
TextBox:	SO THAP PHAN		Chuyen sang THAP	
txtThapPhan txtThapLucPhan	So THAP LUC PH	IAN:	Chuyen sang TH/	AP PHAN
	:			

2. Tạo một Standard Module (file dùng để chứa code dùng chung cho các form trong ứng dụng): Nhấp phải vào Project và chọn Add/Module:



3. Trong file module vừa tạo, nhập code sau vào:

Lưu ý:

- Sinh viên phải suy nghĩ để hiểu chức năng (mục đích) của các hàm/thủ tục bên dưới.
- Chú ý các chú thích (1), (2) và (3) về khai báo toàn cục và cục bộ (Public và Private) để nắm được phạm vi của biến.

Option Explicit

```
'(1) Public: TOÀN CỤC: Biến có thể sử dụng ở mọi nơi trong chương
trình.
Public ThapPhan As Long
Public ThapLucPhan As String
'(2) Private: CỤC BỘ (trong module): Biến chỉ có thể dùng trong
module (file) mà nó được khai báo.
Private CoSo As Integer
Private Function ChuSoThapLuc(ChuSoThapPhan As Integer) As String
'(3) Dim = Private: CỤC BỘ (trong hàm): Biến chỉ có thể dùng
trong hàm mà nó được khai báo.
Dim ChuSoThapLucPhan As String
Select Case ChuSoThapPhan
Case Is <= 9
ChuSoThapLucPhan = Format(ChuSoThapPhan)
```

```
Case 10
        ChuSoThapLucPhan = "A"
    Case 11
        ChuSoThapLucPhan = "B"
    Case 12
        ChuSoThapLucPhan = "C"
    Case 13
        ChuSoThapLucPhan = "D"
    Case 14
        ChuSoThapLucPhan = "E"
    Case 15
        ChuSoThapLucPhan = "F"
    End Select
    ChuSoThapLuc = ChuSoThapLucPhan
End Function
'Trả về giá trị thập phân của chữ số thập lục phân. Ví dụ: A → 10
Private Function GiaTriChuSoThapLucPhan (ChuSoThapLucPhan As
String) As Integer
    Dim ChuSoThapPhan As Integer
    If ChuSoThapLucPhan <= 9 Then</pre>
        GiaTriChuSoThapLucPhan = Val(ChuSoThapLucPhan)
        Exit Function
    End If
    Select Case ChuSoThapLucPhan
   Case "A", "a"
        ChuSoThapPhan = 10
    Case "B", "b"
        ChuSoThapPhan = 11
    Case "C", "c"
        ChuSoThapPhan = 12
    Case "D", "d"
        ChuSoThapPhan = 13
    Case "E", "e"
        ChuSoThapPhan = 14
    Case "F", "f"
        ChuSoThapPhan = 15
    End Select
    GiaTriChuSoThapLucPhan = ChuSoThapPhan
End Function
```

```
Public Function ThapPhanSangThapLucPhan(ThapPhan As Long) As
String
Dim ThapLucPhan_String As String
CoSo = 16
Do
ThapLucPhan_String = ChuSoThapLuc(ThapPhan Mod CoSo) &
ThapLucPhan_String
ThapPhan = ThapPhan \ CoSo
Loop While (ThapPhan <> 0)
ThapPhanSangThapLucPhan = ThapLucPhan_String
End Function
```

Public Function ThapLucPhanSangThapPhan(ThapLucPhan As String) As Long

```
Dim ThapPhan As Long
Dim SoMu As Integer
Dim TrongSo As Integer
CoSo = 16
SoMu = 0
Do While (Len(ThapLucPhan) <> 0)
TrongSo = GiaTriChuSoThapLucPhan(Right(ThapLucPhan, 1))
ThapPhan = ThapPhan + TrongSo * (CoSo ^ SoMu)
ThapLucPhan = Left(ThapLucPhan, Len(ThapLucPhan) - 1)
SoMu = SoMu + 1
Loop
ThapLucPhanSangThapPhan = ThapPhan
```

```
End Function
```

4. Nhấp đúp vào nút *cmdThapPhanSangThapLucPhan* và nhập code:

```
Option Explicit
```

```
Private Sub cmdThapPhanSangThapLucPhan_Click()
```

```
ThapPhan = Val(txtThapPhan.Text)
ThapLucPhan = ThapPhanSangThapLucPhan(ThapPhan)
```

```
txtThapLucPhan.Text = ThapLucPhan
```

```
End Sub
```

5. Nhấp đúp vào nút *cmdThapLucPhanSangThapPhan* và nhập code:

```
Private Sub cmdThapLucPhanSangThapPhan_Click()
    ThapLucPhan = txtThapLucPhan.Text
    ThapPhan = ThapLucPhanSangThapPhan(ThapLucPhan)
    txtThapPhan.Text = Format(ThapPhan)
End Sub
```

🔄. Chuyen doi Thap p	han <-> Thap luc pł	han 🗖 🗖 🗙
So THAP PHAN	48777	Chuyen sang THAP LUC PHAN
So THAP LUC PHAN	BE89	Chuyen sang THAP PHAN

- → *Câu hỏi*: Sinh viên trả lời các câu hỏi sau để *nắm rõ về phạm vi của biến*:
 - Thêm lệnh sau vào thủ tục cmdThapPhanSangThapLucPhan_Click() (trong form):

```
Private Sub cmdThapPhanSangThapLucPhan_Click()
        CoSo = 10 'Lệnh mới thêm vào.
```

••••

→ Khi chạy chương trình gặp lỗi gì? Giải thích tại sao biến CoSo đã được khai báo (ở vị trí (2) trong module) nhưng chương trình vẫn bị lỗi.

Thay đổi khai báo Public ở vị trí (1) thành Private:

Private ThapPhan As Long

→ Khi chạy chương trình gặp lỗi gì? Giải thích tại sao gặp lỗi này.

Đưa khai báo Private CoSo As Integer ở vị trí (2) vào bên trong một hàm, ví dụ đưa vào hàm GiaTriChuSoThapLucPhan():

```
Private Function GiaTriChuSoThapLucPhan (ChuSoThapLucPhan As
String) As Integer
Dim ChuSoThapPhan As Integer
Dim CoSo As Integer 'Khai báo đưa vào từ vị trí (2).
...
```

→ Khi chạy chương trình gặp lỗi gì? Giải thích tại sao gặp lỗi này.

• Thêm lệnh sau vào hàm ThapPhanSangThapLucPhan():

```
Public Function ThapPhanSangThapLucPhan(ThapPhan As Long) As
String
Dim ThapLucPhan_String As String
ChuSoThapLucPhan = "A" 'Lệnh mới thêm vào.
```

•••

→ Khi chạy chương trình gặp lỗi gì? Giải thích tại sao biến ChuSoThapLucPhan đã được khai báo (ở vị trí (3)) nhưng chương trình vẫn bị lỗi.

6.2 TÍNH TỔ HỢP

6.2.1 Mô tả

Tính tổ hợp n chập k theo công thức:

$$\binom{n}{k} = \frac{n!}{k! (n-k)!}$$

Dữ liệu vào: n và k (số nguyên không âm).

Dữ liệu ra: tổ hợp n chập k.

Điều khiển: nút cmdTinh: tính tổ hợp.

6.2.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai13_TinhToHop (tự tạo trước).
- 1. Tạo form như hình (kể từ bài này, tên control (name) được ghi lên trên control):

🖏 Tinh to hop n cl	iap k 📃 🔳	2	×	
Nhap n:	txtN		· · · · · · ·	
Nhap k:	txtK		· · · · · · · · · · · · · · · · · · ·	
cmdTinh	b KQ		· · · · · · · · · · · · · · · · · · ·	
	••••••	: :	::	

2. Tạo một Standard Module và nhập code sau:

```
Option Explicit
```

```
'HÀM tính giai thừa
Private Function GiaiThua(n As Integer) As Long
                                        'Kiểu kết quả trả về: Long
    Dim KQ As Long
    Dim i As Integer
    KQ = 1
    For i = 1 To n Step 1
        KQ = KQ * i
    Next
                      'Trả về kết quả qua tên hàm.
    GiaiThua = KQ
End Function
Public Function ToHop(n As Integer, k As Integer) As Double
    Dim KQ As Double
    KQ = GiaiThua(n) / GiaiThua(k) / GiaiThua(n - k)
    ToHop = KQ
```

```
End Function
```

3. Nhấp đúp vào nút *cmdTinh* và nhập code:

```
Private Sub cmdTinh_Click()
   Dim n As Integer
   Dim k As Integer
   n = Val(txtN.text)
   k = Val(txtK.text)
   lblKQ.Caption = Format(ToHop(n, k))
End Sub
```

4. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả:

5. Tinh to hop n cha	ip k 🗖 🗖 🗙
Nhap n:	9
Nhap k:	2
cmdTinh	36

5. Viết lại phần tính giai thừa dùng *thủ tục*: Mở Module1 và thêm code sau:

```
'THỦ TUC tính giai thừa
Private Sub TinhGiaiThua(n As Integer, KQ As Long)
                                                     'Không trả về
                                                      kết quả qua
                                                      tên thủ tục
    Dim i As Integer
    KQ = 1
    For i = 1 To n Step 1
        KQ = KQ * i
    Next
End Sub
Public Function ToHop DungThuTucTinhGiaiThua(n As Integer, k As
Integer) As Double
    Dim KQ As Double
    Dim GiaiThua n As Long
    Dim GiaiThua k As Long
    Dim GiaiThua n k As Long
    'Cách gọi Thủ tục không dùng từ khóa Call:
    TinhGiaiThua n, GiaiThua n 'Không có dấu ngoặc ()
    TinhGiaiThua k, GiaiThua k
    TinhGiaiThua n - k, GiaiThua n k
```

```
'Hoặc dùng cách gọi có từ khóa Call:
'Call TinhGiaiThua(n, GiaiThua_n) 'Dùng Call, có dấu
ngoặc ()
'Call TinhGiaiThua(k, GiaiThua_k)
'Call TinhGiaiThua(n - k, GiaiThua_n_k)
KQ = GiaiThua_n / GiaiThua_k / GiaiThua_n_k
ToHop_DungThuTucTinhGiaiThua = KQ
End Function
6. Nhấp đúp vào nút cmdTinh và sửa dòng lệnh:
lblKQ.Caption = Format (ToHop(n, k))
```

```
thành:
```

```
lblKQ.Caption = Format(ToHop_DungThuTucTinhGiaiThua(n, k))
```

5. Tinh to hop n chap	pk 🗖 🗖 🗙
Nhap n:	8
Nhap k:	3
cmdTinh	56

→ *Câu hỏi:* Kết quả 2 lần chạy có khác nhau không?

→ *Yêu cầu:* Sinh viên xem lại code lần nữa để thấy được *sự khác nhau trong cách viết và sử dụng* hàm và thủ tục (function và sub).

6.3 THAM TRỊ VÀ THAM CHIẾU

6.3.1 Mô tả

Dữ liệu vào: 2 số nguyên A và B.

Dữ liệu ra: A được cộng 4, B nhân 3 hoặc ngược lại.

Điều khiển: 2 nút cmd_ANhan4_BCong3 và cmd_ACong3_BNhan4: chạy thủ tục tính A, B tương ứng.

6.3.2 Thực hiện

0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai14_ThamTriVaThamChieu (tự tạo trước).

1. Tao form như hình:



2. Nhấp đúp vào nút *cmd_ANhan4_BCong3* và nhập code (nhập thêm 2 thủ tục **Nhan4** và **Cong3** vào bên trên thủ tục cmd_ANhan4_BCong3_Click):

```
Private Sub Nhan4 (ByVal A As Integer)
       A = A \star 4
   End Sub
   Private Sub Cong3(A As Integer)
       A = A + 3
   End Sub
   Private Sub cmd ANhan4 BCong3 Click()
       Dim A As Integer
       Dim B As Integer
       A = Val(txtA.Text)
       B = Val(txtB.Text)
       Call Nhan4(A)
       Call Cong3(B)
       lblA.Caption = A
       lblB.Caption = B
   End Sub
3. Nhấp đúp vào nút cmd_ACong3_BNhan4 và nhập code:
```

```
Private Sub cmd ACong3 BNhan4 Click()
    Dim A As Integer
    Dim B As Integer
    A = Val(txtA.Text)
    B = Val(txtB.Text)
    Call Cong3(A)
    Call Nhan4(B)
    lblA.Caption = A
    lblB.Caption = B
End Sub
```

- 4. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả:
- → Lưu ý: Sinh viên tự tính toán giá trị mới của A và B trước khi chạy chương trình. Giải thích tại sao 2 thủ tục cho kết quả khác nhau (đọc lại phần Tham trị và tham chiếu trong mục 4. Kiến thức bổ trợ nếu chưa rõ).

6	Tham tr	i va tham chieu		
	А	5	В	1
	cmd_AN	han4_BCong3	cmd_AC	Cong3_BNhan4
L				

6.4 SẮP XẾP MẢNG TĂNG DẦN

6.4.1 Mô tả

Dữ liệu vào: mảng các số.

Dữ liệu ra: mảng các số được sắp tăng dần.

Điều khiển: nút cmdSap: thực hiện sắp xếp mảng.

6.4.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai15_SapXepMangTang (tự tạo trước).
- 1. Tạo form như hình:

🖏 Sap xep r	nang tang dan	
Mang	txt1 txt2 txt3	txt4 txt5
cmdSap	lpiko	· · · · · · · · · · · · · · · · · · ·

2. Nhấp đúp vào nút *cmdSap* và nhập code:

Const N As Integer = 4

```
Private Sub cmdSap_Click()
```

```
Dim a(N) As Integer
Dim i As Integer, j As Integer
a(0) = Val(txt1.Text)
a(1) = Val(txt2.Text)
a(2) = Val(txt3.Text)
a(3) = Val(txt4.Text)
a(4) = Val(txt5.Text)
```

```
For i = 0 To N - 1
        For j = i + 1 To N
            If a(i) > a(j) Then
                HoanDoiViTri a(i), a(j)
            End If
        Next
    Next
    lblKQ.Caption = a(0) & " " & a(1) & " " & a(2) & " " & a(3) &
" " & a(4)
End Sub
Private Sub HoanDoiViTri (a As Integer, b As Integer)
    Dim temp As Integer
    temp = a
    a = b
    b = temp
End Sub
```

🖪, Sap xep ma	ang tang dan			- 0	x
Mang	4	2	-6	7	5
[cmdSap]	-62457				

4. Sửa dòng khai báo thủ tục HoanDoiViTri():

```
Private Sub HoanDoiViTri(ByVal a As Integer, ByVal b As Integer)
Dim temp As Integer
temp = a
a = b
b = temp
End Sub
```

- 5. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả.
- → Câu hỏi: Mảng có được sắp xếp không? Tại sao?

6.5 BÂY LÕI BẰNG ON ERROR

6.5.1 Mô tả

Dữ liệu vào: một số nguyên không âm N.

Dữ liệu ra: giai thừa của N.

Điều khiển: nút cmdTinhGiaiThua: thực hiện tính giai thừa.

6.5.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai16_BayLoiBangOnError (tự tạo trước).
- 1. Tạo form như hình:



2. Nhấp đúp vào nút *cmdTinhGiaiThua* và nhập code:

```
Private Sub cmdTinhGiaiThua_Click()
    Dim KQ As Long
    Dim i As Integer
    Dim n As Integer
    n = Val(txtN.Text)
    KQ = 1
    For i = 1 To n Step 1
        KQ = KQ * i
    Next
    lblKQ.Caption = KQ
End Sub
```

 Lưu và nhấn nút Start hay F5 để chạy và xem kết quả: Khi nhập N nhỏ hơn hoặc bằng 12:

🔄, Form1		
Nhap	N: 12	
	cmdTinhGiaiThua	
	479001600	

Khi nhập N lớn hơn 12, một thông báo lỗi xuất hiện và chương trình bị thoát:

ſ	5 Form1			x	
	Nhap N:	13			
		cmdTinhGiai	Thua]	
Microso	ft Visual Basic	100			
Run-tin	ne error '6':				
Overflo	w				
	tione I I I		Dahua		ь I
Con		ina _	Debug	He	aib

→ Để giải quyết vấn đề này, ta dùng cú pháp bẫy lỗi On Error:

4. Nhấp đúp vào nút *cmdTinhGiaiThua* và sửa code:

```
Private Sub cmdTinhGiaiThua Click()
    Dim KQ As Long
    Dim i As Integer
    Dim n As Integer
    n = Val(txtN.Text)
                            'Lệnh bẫy lỗi phải đặt trên những câu
    On Error GoTo XuLyLoi
                             lệnh có thể gây lỗi.
    KQ = 1
    For i = 1 To n Step 1
        KQ = KQ * i
    Next
    lblKQ.Caption = KQ
                'Lệnh thoát khỏi thủ tục, để tránh chạy code xử
   Exit Sub
                 lý lỗi nếu chương trình không có lỗi.
                 (Chú ý: Exit Sub ≠ End Sub)
XuLyLoi:
            'Phần code xử lý lỗi, thường đặt ở cuối hàm/thủ tục,
             và thường dùng Select Case
    Select Case Err.Number
                             ' Err.Number: mã số lỗi
      Case 6
                    ' Lỗi 6: tràn bộ nhớ.
         lblKQ.Caption = "Xin nhap N nho hon."
                   ' Xử lý các lỗi khác.
      Case Else
         lblKQ.Caption = "Mot loi chua xac dinh da xay ra!"
    End Select
```

- End Sub
- 5. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả:

Khi nhập N lớn hơn 12, chương trình không bị thoát:

🔄 Form1	
Nhap N:	13
cma	dTinhGiaiThua
Xin nh	nap N nho hon.

6.6 XÓA KHOẢNG TRẮNG THỪA TRONG CHUỗI

6.6.1 Mô tả

Dữ liệu vào: một chuỗi có các khoảng trắng thừa, ví dụ: "Su Pham Ky Thuat TP.HCM". Dữ liệu ra: chuỗi đã được xóa các khoảng trắng thừa: "Su Pham Ky Thuat TP.HCM".

Điều khiển: nút cmdXoaKhoangTrangThua: thực hiện xóa các khoảng trắng thừa của chuỗi.

Yêu cầu: Chỉ được phép sử dụng hàm *Len()* và *Mid()*. Không sử dụng các hàm và các thủ tục xử lý chuỗi có sẵn khác của VB.

6.6.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai17_XoaKhoangTrangThuaTrongChuoi (tự tạo trước).
- 1. Tạo form như hình:

5, Form1	
Chuoi:	txtChuoi
	cmdXoaKhoangTrangThua
	IbIKQ

2. Nhấp đúp vào nút *cmdXoaKhoangTrangThua* và nhập code:

```
Private Sub cmdXoaKhoangTrangThua Click()
    Dim KQ As String
    Dim i As Integer
    KO = txtChuoi.Text
    'Xóa khoảng trắng ở đầu chuỗi.
    Do While Mid(KQ, 1, 1) = " "
        KQ = Mid(KQ, 2)
    Loop
    If Len(KO) > 0 Then
        'Xóa khoảng trắng ở cuối chuỗi.
        Do While Mid(KQ, Len(KQ), 1) = " "
            KQ = Mid(KQ, 1, Len(KQ) - 1)
        qool
        'Xóa khoảng trắng thừa bên trong chuỗi, bằng cách duyệt
        từ đầu đến hết chuỗi, nếu gặp 2 khoảng trắng liên tiếp
        thì xóa bớt 1 khoảng trắng.
        For i = 1 To Len(KQ) Step 1
```

5, Form1	
Chuoi:	Su Pham Ky Thuat TP.HCM
	cmdXoaKhoangTrangThua
	<su ky="" pham="" thuat="" tp.hcm=""></su>

7 Bài tập đề nghị

7.1 Viết chương trình chuyển đổi số giữa các hệ đếm: nhị phân <-> thập phân, và nhị phân <-> thập lục phân.

7.2 Viết chương trình đếm số từ trong chuỗi

Mô tả: Đếm số từ trong chuỗi, ví dụ: chuỗi "Su Pham Ky Thuat TP.HCM" có 5 từ.

Gọi ý: Dùng thủ tục xóa khoảng trắng thừa trong chuỗi (bài **6.6** X**óa khoảng trắng thừa trong chuỗi**). Sau đó đếm số khoảng trắng trong chuỗi. Số từ = Số khoảng trắng + 1.

7.3 Viết chương trình sắp xếp mảng tăng dần, với mảng là một chuỗi các số ngăn cách bằng khoảng trắng*

Mô tả: Người dùng sẽ nhập mảng bằng một chuỗi, ví dụ, "2 -3 12 4 5 77". Sau đó chương trình sẽ tách chuỗi thành các số và đưa vào mảng rồi thực hiện sắp xếp.

🖪, Sap xep ma	ng tang dan	x
Mang	3 0 4 -6 7 8 77 0 6 -6 3	
cmdSap	-6-60033467877	

Gọi ý: Dùng *mảng động*. Số phần tử của mảng là số từ trong chuỗi (bài **7.2**). Cắt chuỗi thành các số dựa vào khoảng trắng ngăn cách và đưa vào mảng để sắp xếp.

