
LẬP TRÌNH VISUAL BASIC

BÀI TẬP THỰC HÀNH SỐ 4

CẤU TRÚC ĐIỀU KHIỂN

1. Chuẩn đầu ra.

Sau khi học xong chương này, sinh viên có thể:

- Trình bày được cú pháp các cấu trúc điều khiển.
- Chuyển đổi qua lại được giữa các cấu trúc lặp sử dụng For ... Next, Do While ... Loop, Do ... Loop Until.
- Viết được chương trình cơ bản có sử dụng các cấu trúc điều khiển.
- Thái độ làm việc chăm chỉ, hình thành phong cách lập trình.

2. Chuẩn bị

SV phải làm (trên giấy) các bài tập về mảng đã làm trên lớp giờ lý thuyết và làm thêm ở nhà.

3. Phương tiện

Phòng máy có cài chương trình Microsoft Visual Basic 6.0

4. Thời lượng

4 tiết.

5. Tóm tắt lý thuyết

Cấu trúc điều khiển được chia làm ba loại: cấu trúc tuần tự, cấu trúc chọn, cấu trúc lặp.

- Cấu trúc tuần tự: chương trình được thực hiện tuần tự từ trên xuống dưới, dòng lệnh nào viết trước thì thực hiện trước.
- Cấu trúc chọn: cho phép thực hiện những công việc khác nhau tùy theo điều kiện nào đó.

- Cấu trúc **If dạng khuyết**

If <Điều kiện> **Then** <Lệnh>

If <Điều kiện> **Then**

<Nhóm lệnh>

End If

- Cấu trúc **If dạng đủ**

If <Điều kiện> **then** <Lệnh 1> **else** <Lệnh 2>

If <Điều kiện> **Then**

<Nhóm lệnh 1>

Else

<Nhóm lệnh 2>

End If

- Cấu trúc **If dạng phức tạp**

If <Điều kiện 1> **then**

 <Nhóm lệnh 1>

ElseIf <Điều kiện 2> **then**

 <Nhóm lệnh 2>

ElseIf <Điều kiện n> **then**

 <Nhóm lệnh n>

Else

 <Nhóm lệnh n + 1>

EndIf

- Cấu trúc **Select Case dạng khuyết**

Select Case <Biểu thức>

Case <Giá trị 1>

 <Nhóm lệnh 1>

Case <Giá trị 2>

 <Nhóm lệnh 2>

 ...

Case <Giá trị n>

 <Nhóm lệnh n>

End Select

- Cấu trúc **Select Case dạng đủ**

Select Case <Biểu thức>

Case <Giá trị 1>

 <Nhóm lệnh 1>

Case <Giá trị 2>

 <Nhóm lệnh 2>

...

Case <Giá trị n>

<Nhóm lệnh n>

Case Else

<Nhóm lệnh n + 1>

End Select

- Cấu trúc lặp:

○ Cấu trúc lặp xác định **For ... Next**

For <Biến đếm> = <Trị đầu> **to** <Trị cuối> [**Step** <Bước tăng>]

<Lệnh>

Next [<Biến đếm>]

Chức năng: <Lệnh> được thực hiện từ <Trị đầu> đến <Trị cuối>, giá trị của <Biến đếm> được thay đổi theo <Bước tăng>.

○ Cấu trúc lặp không xác định **Do ... Loop**

Dạng 1:

Do While <Điều kiện>

<Nhóm lệnh>

Loop

Chức năng: nếu <Điều kiện> đúng thì <Nhóm lệnh> được thực thi và lặp lại cho đến khi nào <Điều kiện> sai.

Dạng 2:

Do Until <Điều kiện>

<Nhóm lệnh>

Loop

Chức năng: nếu <Điều kiện> sai thì <Nhóm lệnh> được thực thi và lặp lại cho đến khi nào <Điều kiện> đúng.

Dạng 3:

Do

<Nhóm lệnh>

Loop While <Điều kiện>

Chức năng: <Nhóm lệnh> được thực thi và lặp lại trong khi <Điều kiện> đúng.

Dạng 4:

Do

<Nhóm lệnh>

Loop Until <Điều kiện>

Chức năng: <Nhóm lệnh> được thực thi và lặp lại trong khi <Điều kiện> sai.

6. Nội dung thực hành

- 6.1 Viết chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$.
- 6.2 Viết chương trình tính số ngày trong tháng của năm.
 - Nếu tháng = 4, 6, 9, 11 thì số ngày = 30
 - Nếu tháng = 1, 3, 5, 7, 8, 10, 12 thì số ngày = 31
 - Nếu tháng = 2 thì số ngày = 28 hoặc 29 (nếu là năm nhuận)
 - Năm nhuận là năm chia hết cho 400 hoặc là năm chia hết cho 4 và không chia hết cho 100.
- 6.3 Viết chương trình tính tiền điện phải trả theo số kWh sử dụng được nhập vào. Bảng tính tiền điện được cho như sau:

Mức sử dụng	Giá bán điện (đồng/kWh)
Cho kWh từ 1 - 100	1284
Cho kWh từ 101 - 150	1457
Cho kWh từ 151 - 200	1843
Cho kWh từ 201 - 300	1997
Cho kWh từ 301 - 400	2137
Cho kWh từ 401 trở lên	2192

Ví dụ: nếu sử dụng 223 kWh thì số tiền phải trả là:

$$100 \cdot 1284 + 50 \cdot 1457 + 50 \cdot 1843 + 23 \cdot 1997 = 339331$$

- 6.4 Cho số nguyên n , tính $s = n! = 1 \times 2 \times \dots \times n$, với $n \geq 0$ ($0! = 1$).
- 6.5 Cho số nguyên n , tính tổng $s = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + L + \frac{(-1)^n x^{2n+1}}{(2n+1)!}$, với $n \geq 0$
- 6.6 Cho số nguyên dương n , tìm số đảo ngược của n . Ví dụ, với $n = 12345$ thì số đảo ngược của n là 54321.
- 6.7 Cho số nguyên n , cho biết n có phải là số nguyên tố hay không?
- 6.8 Viết chương trình tính ước số chung lớn nhất của hai số nguyên dương.
- 6.9 Viết chương trình kiểm tra số nguyên dương n có phải là số hoàn hảo hay không?

6.10 Viết chương trình kiểm tra số nguyên dương n có phải là số chính phương hay không?

7. Bài tập thêm

7.1 Viết chương trình nhập vào tử số và mẫu số của một phân số, thực hiện tối giản phân số đó.

7.2 Viết chương trình xếp loại học lực theo điểm trung bình được nhập vào.

- $\text{DiemTB} < 5.0 \rightarrow \text{Yếu}$
- $5.0 \leq \text{DiemTB} < 7.0 \rightarrow \text{Trung bình}$
- $7.0 \leq \text{DiemTB} < 8.0 \rightarrow \text{Khá}$
- $8.0 \leq \text{DiemTB} < 9.0 \rightarrow \text{Giỏi}$
- $\text{DiemTB} > 9.0 \rightarrow \text{Xuất sắc}$

7.3 Viết chương trình tính tổng các ước số của một số nguyên dương n .

7.4 Viết chương trình in ra n số nguyên tố đầu tiên.

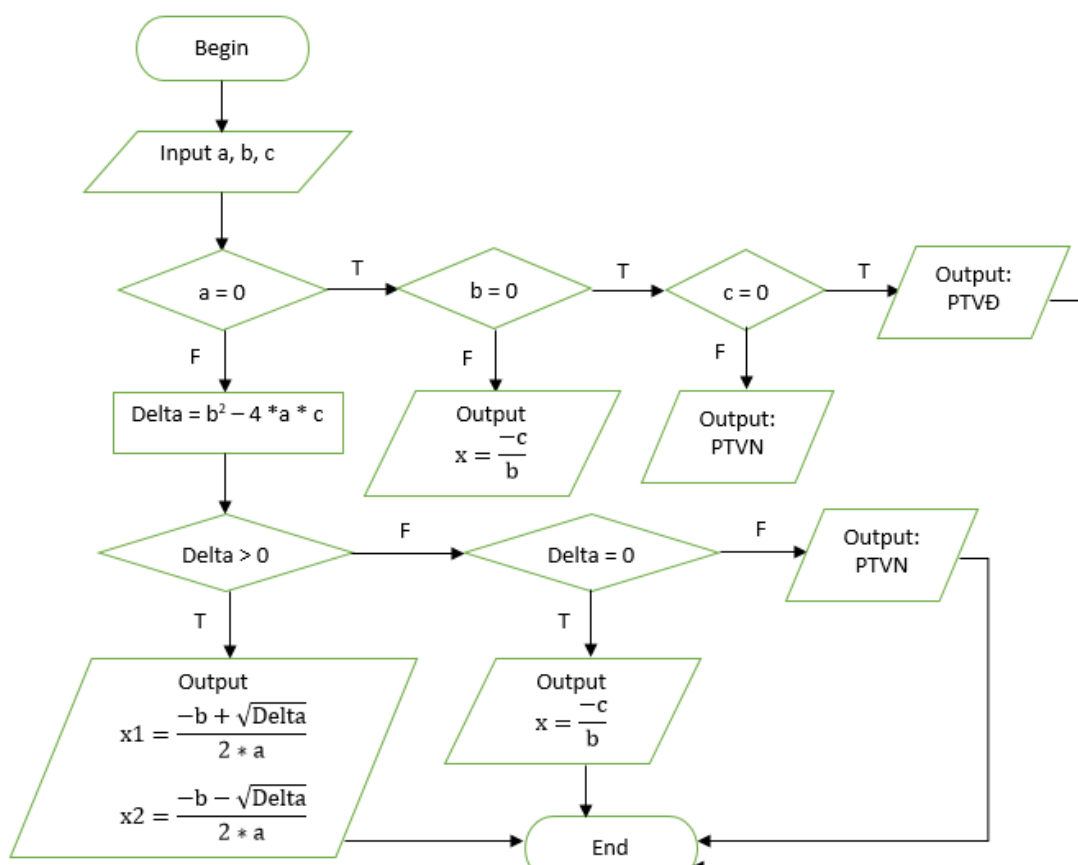
7.5 Viết chương trình tính số Fibonacci thứ n . công thức truy hồi của dãy Fibonacci là:

$$F_n := F(n) := \begin{cases} 0, & \text{khi } n = 0; \\ 1, & \text{khi } n = 1; \\ F(n-1) + F(n-2) & \text{khi } n > 1. \end{cases}$$

8. Hướng dẫn thực hành:

Bài 6.1: Viết chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$

Mô tả thuật toán



Cài đặt thuật toán

Dùng cấu trúc **If dạng đủ và dạng phức tạp**

If $a = 0$ **then**

If $b = 0$ **then**

If $c = 0$ **then**

Print “Phương trình vô định”

Else

Print “Phương trình vô nghiệm”

End If

Else

Print $x = \frac{-c}{b}$

End If

Else

$\Delta = b^2 - 4*a*c$

If $\Delta > 0$ **then**

Print $x1 = \frac{-b + \sqrt{\Delta}}{2*a}, x2 = \frac{-b - \sqrt{\Delta}}{2*a}$

Else If $\Delta < 0$ **then**

Print “Phương trình vô nghiệm”

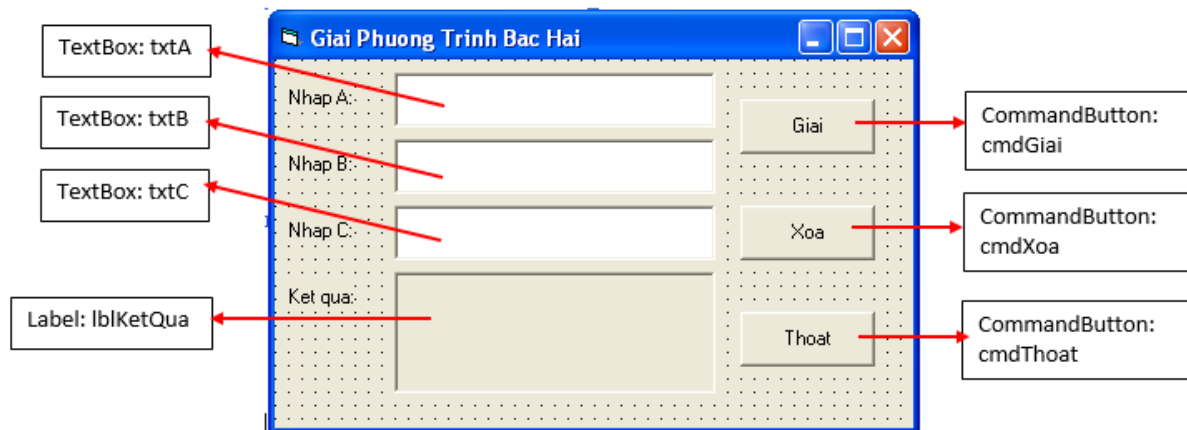
Else

Print $x1 = x2 = \frac{-b}{2*a}$

End If

Thiết kế chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới



```

Private Sub cmdGiai_Click()
    'Khai bao bien
    Dim a As Double, b As Double, c As Double
    Dim x1 As Double, x2 As Double
    Dim Delta As Double

    'Lay du lieu
    a = Val(txtA.Text)
    b = Val(txtB.Text)
    c = Val(txtC.Text)

    'Xu ly
    If a = 0 Then
        If b = 0 Then
            If c = 0 Then
                lblKQ.Caption = "PT vo dinh"
            Else 'c khac 0
                lblKQ.Caption = "PT vo nghiem"
            End If
        Else 'b khac 0
            lblKQ.Caption = "PT co nghiem X = " & (-c / b)
        End If
    Else 'a khac 0
        Delta = b * b - 4 * a * c
        If Delta = 0 Then
            lblKQ.Caption = "PT co nghiem kep X= " & (-b / (2 * a))
        ElseIf Delta < 0 Then lblKQ.Caption = "PT vo nghiem"
        Else
            x1 = (-b + Sqr(Delta)) / (2 * a)
            x2 = (-b - Sqr(Delta)) / (2 * a)
            lblKQ.Caption = "PT co 2 nghiem: X1 = " & x1 & _
                " va X2 = " & x2
        End If
    End If
End Sub

```

```

Private Sub cmdThoat_Click()
    End
End Sub

```

```

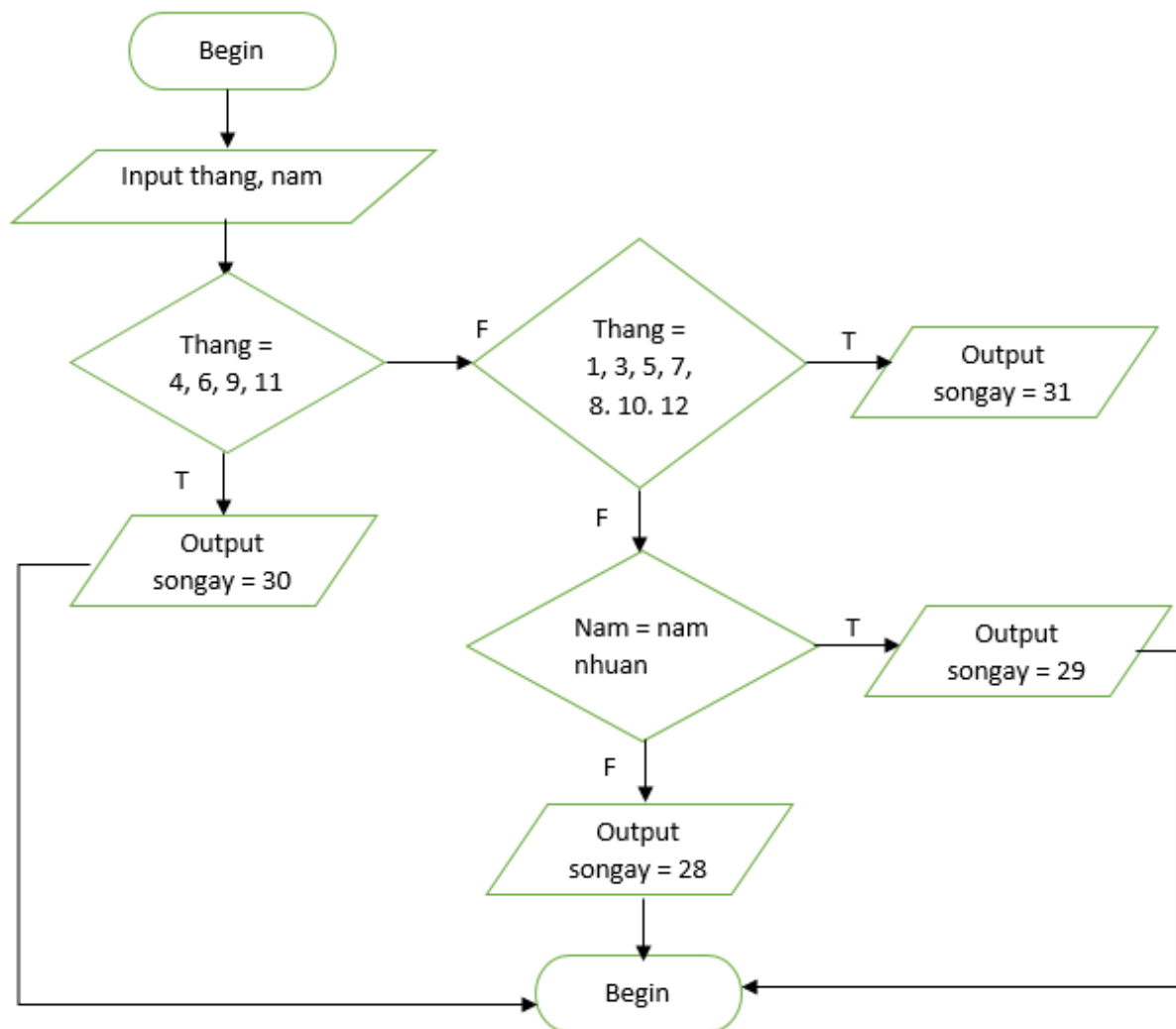
Private Sub cmdXoa_Click()
    txtA.Text = ""
    txtB.Text = ""
    txtC.Text = ""
    txtA.SetFocus

    lblKQ.Caption = ""
End Sub

```

Bài 6.2: Viết chương trình tính số ngày trong tháng của năm

Mô tả thuật toán



Cài đặt thuật toán

Dùng cấu trúc **If phức tạp** và cấu trúc **If đầy đủ**

If thang = 4 Or thang = 6 Or thang = 9 Or thang = 11 **then**

Songay = 30

ElseIf thang = 1 Or thang = 3 Or thang = 5 Or thang = 7 Or thang = 8 Or thang = 10
Or thang = 12 **then**

Songay = 31

Else

If (nam mod 400 = 0) Or ((nam mod 4 = 0) And (nam mod 100 <> 0)) **then**

Songay = 29

Else

Songay = 28

End If

End If

Dùng cấu trúc **Select ... Case** dạng đầy đủ và **If** dạng đầy đủ

Select Case thang

Case 4, 6, 9, 11

Songay = 30

Case 1, 3, 5, 7, 8, 10, 12

Songay = 31

Case Else

If (nam mod 400 = 0) Or ((nam mod 4 = 0) And (nam mod 100 = 0))
then

Songay = 29

Else

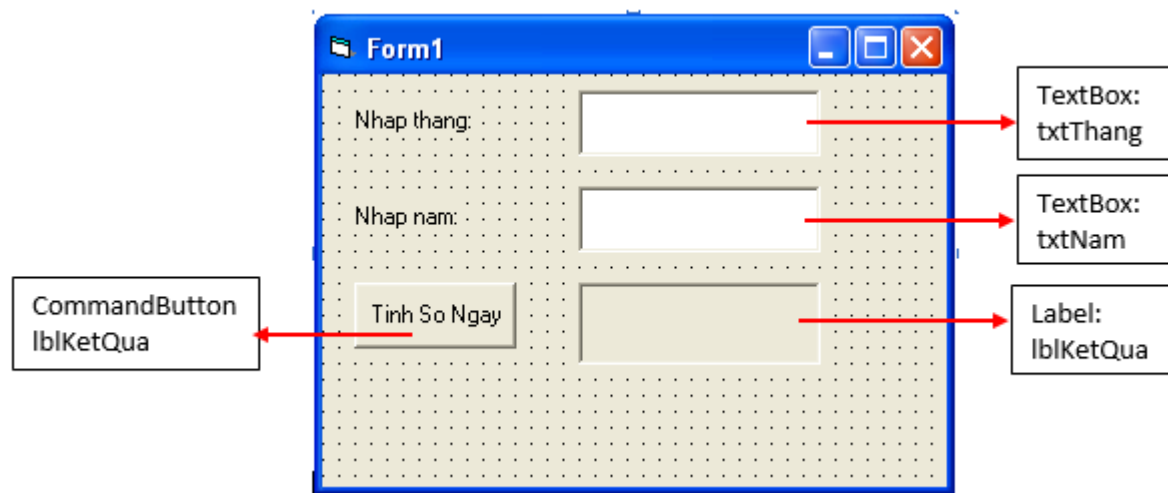
Songay = 28

End If

End Select

Thiết kế chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới



Dùng cấu trúc **If dạng phức tạp** và cấu trúc **If dạng đầy đủ**

```
Private Sub cmdTinhSoNgay_Click()
    'Khai bao bien
    Dim thang As Byte, nam As Integer, songay As Byte

    'Lay du lieu
    thang = Val(txtThang.Text)
    nam = Val(txtNam.Text)

    'Xu ly
    If thang = 4 Or thang = 6 Or thang = 9 Or thang = 11 Then
        songay = 30
    ElseIf thang = 1 Or thang = 3 Or thang = 5 Or thang = 7 Or thang = 8 Or thang = 10 Or thang = 12 Then
        songay = 31
    Else
        If (nam Mod 400 = 0) Or ((nam Mod 4 = 0) And (nam Mod 100 <> 0)) Then
            songay = 29
        Else
            songay = 28
        End If
    End If

    'Xuat ket qua
    lblKetQua.Caption = songay
End Sub
```

Dùng cấu trúc **Select ... Case** và cấu trúc **If dạng đầy đủ**

```
Private Sub cmdTinhSoNgay_Click()
    'Khai bao bien
    Dim thang As Byte, nam As Integer, songay As Byte

    'Lay du lieu
    thang = Val(txtThang.Text)
    nam = Val(txtNam.Text)

    'Xu ly
    Select Case thang
        Case 4, 6, 9, 11
            songay = 30
        Case 1, 3, 5, 7, 8, 10, 12
            songay = 31
        Case Else
            If (nam Mod 400 = 0) Or ((nam Mod 4 = 0) And (nam Mod 100 <> 0)) Then
                songay = 29
            Else
                songay = 28
            End If
        End Select

    'Xuat ket qua
    lblKetQua.Caption = songay
End Sub
```

Bài 6.4: Cho số nguyên n , tính $s = n! = 1 \times 2 \times \dots \times n$, với $n \geq 0$ ($0! = 1$).

Phân tích

Ta cần chú ý rằng đơn vị tính toán số học (ALU) của CPU chỉ có thể nhân hai số tại một thời điểm. Theo định nghĩa trên, ta có:

$$\begin{aligned}0! &= 1 \\1! &= 1 \times 1 \\2! &= 1 \times 2 \\3! &= 1 \times 2 \times 3 \\4! &= 1 \times 2 \times 3 \times 4\end{aligned}$$

Ta thấy $4!$ chứa tất cả các thừa số của $3!$ và chứa thêm 4 nên nó có thể được tính bằng $4 \times 3!$. Ta có thể tổng quát nhận xét này lên là $n!$ có thể được tính bằng $n \times (n-1)!$.

Dùng định nghĩa này ta có thể viết lại

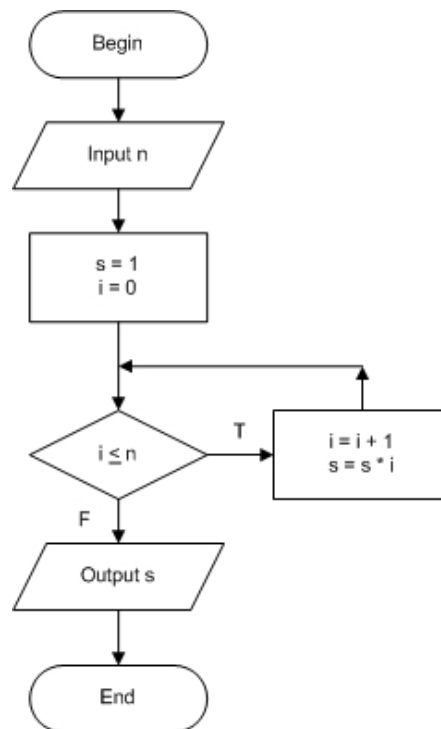
$$\begin{aligned}0! &= 1 \\1! &= 1 \times 0! \\2! &= 2 \times 1! \\3! &= 3 \times 2! \\4! &= 4 \times 3!\end{aligned}$$

Ta có thể bắt đầu với $s = 0! = 1$ và tính $n!$ như bên dưới

$$\begin{array}{ccc} s = 1 & (1) & = 0! \\ \left. \begin{array}{l} s = s \times 1 \\ s = s \times 2 \\ s = s \times 3 \\ s = s \times 4 \\ \vdots \\ M \end{array} \right\} & (2 \dots n+1) & \left\{ \begin{array}{l} = 1! \\ = 2! \\ = 3! \\ = 4! \\ \vdots \\ M \end{array} \right.\end{array}$$

Từ bước 2 trở đi ta lặp lại một quá trình giống nhau. Ở bước thứ $i+1$ ta có $s = s \times i$. Quá trình tổng quát này có thể dùng vòng lặp để biểu diễn. Như vậy để nhân nhiều số ta lặp lại nhiều lần nhân 2 số.

Mô tả thuật toán

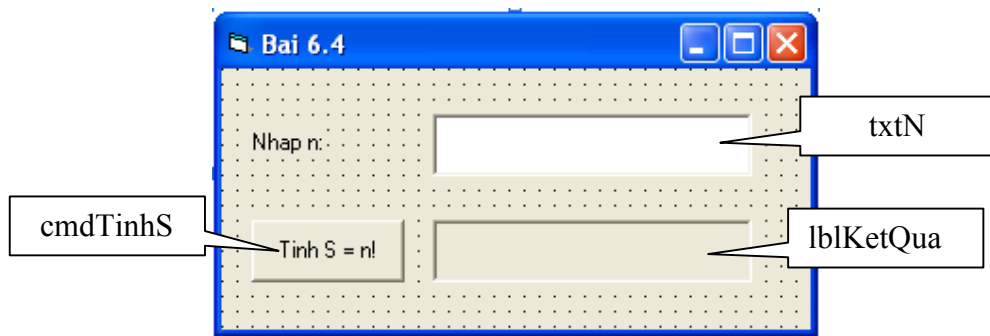


Cài đặt thuật toán

Dùng For	Dùng Do While ... Loop	Dùng Do ... Loop While ...
<pre> s = 1 For i = 1 To n s = s * i Next i Print s </pre>	<pre> s = 1 i = 0 Do While i <= n i = i + 1 s = s * i Loop Print s </pre>	<pre> s = 1 i = 0 Do i = i + 1 s = s * i Loop While i <= n Print s </pre>
Dùng While ... Wend	Dùng Do Until ... Loop	Dùng Do ... Loop Until ...
<pre> s = 1 i = 0 While i <= n i = i + 1 s = s * i Wend Print s </pre>	<pre> s = 1 i = 0 Do Until i > n i = i + 1 s = s * i Loop Print s </pre>	<pre> s = 1 i = 0 Do i = i + 1 s = s * i Loop Until i > n Print s </pre>

Thiết kế chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới.



Mã chương trình như bên dưới

```
Private Sub cmdTinhS_Click()
    'Khai bao bien
    Dim n As Integer
    Dim i As Integer
    Dim s As Integer

    'Lay du lieu
    n = Val(txtN.Text)
    s = 1
    i = 1

    'Xu ly
    Do While i <= n
        s = s * i
        i = i + 1
    Loop

    'Xuat ket qua
    lblKetQua.Caption = s
End Sub
```

Bài 6.5: Cho số nguyên n , tính tổng $s = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n x^{2n+1}}{(2n+1)!}$, với $n \geq 0$

Phân tích

Thành phần quan trọng trong từng số hạng của tổng trên có dạng $\frac{x^i}{i!} = \frac{x}{1} \cdot \frac{x}{2} \cdot \dots \cdot \frac{x}{i}$ có thể được tính độc lập theo kỹ thuật của bài tính $n!$

```
t = 1
For j = 1 To i
    t = t * x / j
Next i
```

Tuy nhiên, để hiệu quả ta muốn tính số hạng thứ i dựa vào số hạng thứ $i - 1$ (giống trong bài tính $n!$ ở trên, ta tính $i!$ dựa vào $(i - 1)!$). Ta nhận thấy

$$\frac{(-1)^i x^{2i+1}}{(2i+1)!} = -\frac{x^2}{2i(2i+1)} \cdot \frac{(-1)^{i-1} x^{2i-1}}{(2i-1)!}$$

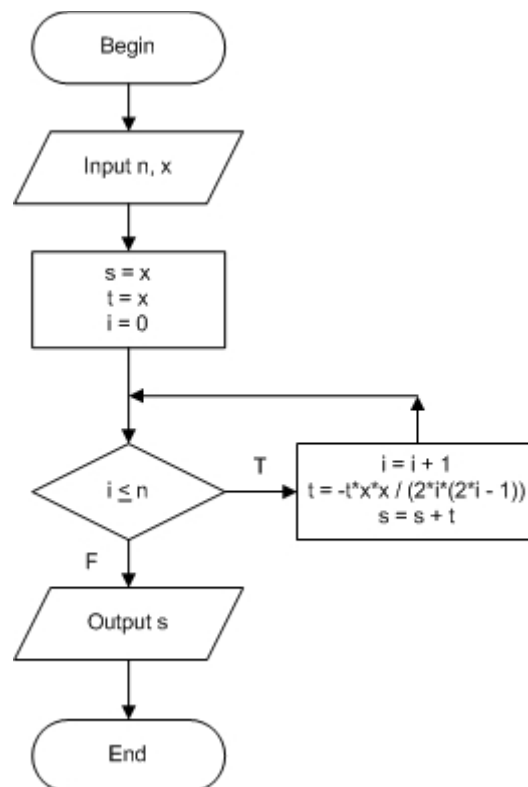
Ta có thể bắt đầu với

$$s = x, t = x, i = 1$$

Số hạng thứ i và tổng có thể được tính trực tiếp từ các số hạng trước nó.

$$\begin{cases} i = i + 1 \\ t = -\frac{tx^2}{2i(2i+1)} \\ s = s + t \end{cases}$$

Mô tả thuật toán

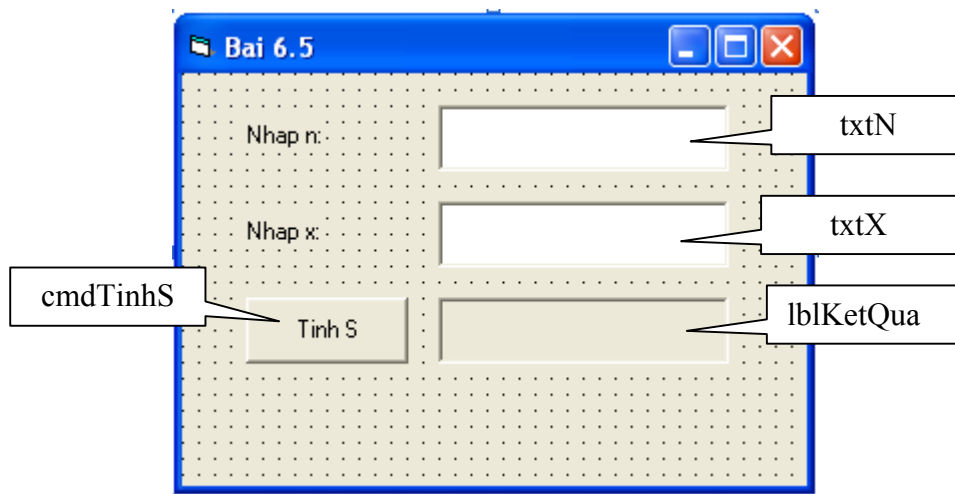


Cài đặt thuật toán

Dùng For	Dùng Do While ... Loop	Dùng Do ... Loop Until ...
<pre> s = x t = x For i = 1 To n t = -t*x*x / (2*i*(2*i-1)) s = s + t Next i Print s </pre>	<pre> s = x t = x i = 0 Do While i <= n i = i + 1 t = -t*x*x / (2*i*(2*i-1)) s = s + t Loop Print s </pre>	<pre> s = x t = x i = 0 Do i = i + 1 t = -t*x*x / (2*i*(2*i-1)) s = s + t Loop Until i > n Print s </pre>

Thiết kế chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới.



Mã chương trình như bên dưới

```
Private Sub cmdTinhS_Click()  
    'Khai bao bien  
    Dim n As Integer  
    Dim x As Double  
    Dim t As Double  
    Dim s As Double  
    Dim i As Integer  
  
    'Lay du lieu  
    n = Val(txtN.Text)  
    x = Val(txtX.Text)  
  
    'Xu ly  
    s = x  
    t = x  
    i = 0  
    Do While i <= n  
        i = i + 1  
        t = -t * x * x / (2 * i * (2 * i - 1))  
    Loop  
  
    'Xuat ket qua  
    lblKetQua.Caption = s  
End Sub
```

Bai 6.6: Cho số nguyên dương n , tìm số đảo ngược của n . Ví dụ, với $n = 12345$ thì số đảo ngược của n là 54321.

Phân tích

Số 12345 thật sự được phân tích như sau

$$12345 = 1 \cdot 10^4 + 2 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

Để làm được điều này ta phải lấy được từng chữ số của n. Vấn đề là ta nên thực hiện điều này từ trái sang phải hay ngược lại? Bởi vì ta không biết được n có bao nhiêu chữ số, sẽ dễ hơn nếu ta thực hiện điều này từ phải sang trái, nghĩa là xác định chữ số ở hàng nhỏ nhất rồi nhỏ nhì, ... Với số $n = 12345$, ta sẽ xác định và lấy số 5 ra trước, số còn lại sẽ là 1234.

Ta có thể lấy số 5 và thu được số 1234 bằng hai thao tác

$$12345 \bmod 10 \rightarrow 5, \text{ và}$$

$$12345 / 10 \rightarrow 1234$$

Vì vậy ta áp dụng 2 phép tính sau

$$r = n \bmod 10$$

$$n = n / 10$$

Lặp lại quá trình giống như thế này cho giá trị n mới ta sẽ thu được từng chữ số của n.

Sau khi lấy được từng chữ số của n, bước tiếp theo là đảo ngược các chữ số, nghĩa là số lấy ra trước (số ở hàng nhỏ hơn) sẽ được viết trước. Ví dụ, khi ta lấy ra số 5 trước rồi đến số 4 (45) thì kết quả số 5 sẽ đứng trước số 4 (54). Số ban đầu 45 có thể được đảo ngược bằng cách lấy ra số 5, nhân nó với 10 rồi cộng nó với 4. Cách làm này có thể được áp dụng cho số có nhiều chữ số hơn. Chẳng hạn, 3 số cuối 345 sẽ được đảo ngược thành 543, sau khi lấy được số 5 và số 4 ta đã thu được 54, khi lấy thêm số 3, ta sẽ thu được 543 bằng cách lấy 54 nhân với 10 và cộng với 3.

Quá trình này có thể được minh họa như bên dưới. Giả sử t là số đảo ngược, giá trị khởi đầu của t là 0.

$$\text{Đầu tiên lấy được số 5 thì giá trị của t sẽ là } t = t * 10 + 5 = 0 * 10 + 5 = 5$$

$$\text{Lấy được số 4: } t = t * 10 + 4 = 5 * 10 + 4 = 54$$

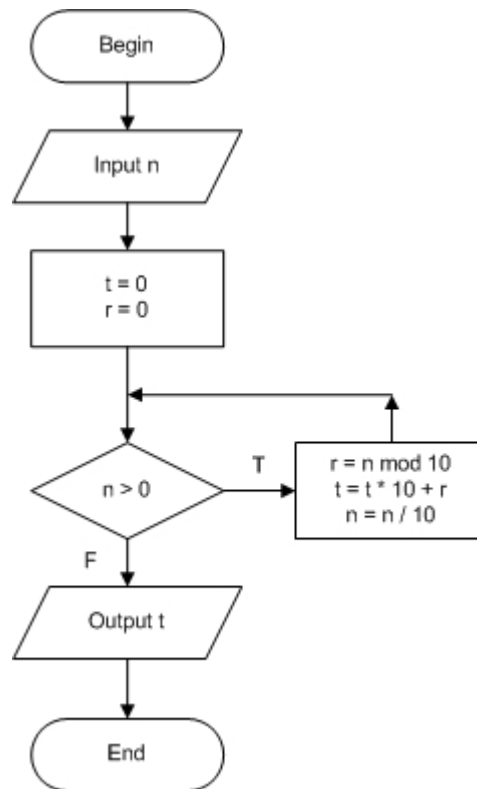
$$\text{Lấy được số 3: } t = t * 10 + 3 = 54 * 10 + 3 = 543$$

$$\text{Lấy được số 2: } t = t * 10 + 2 = 543 * 10 + 2 = 5432$$

$$\text{Lấy được số 1: } t = t * 10 + 1 = 5432 * 10 + 1 = 54321$$

n	r	t	Bước
12345	5	5	1
1234	4	54	2
123	3	543	3
12	2	5432	4
1	1	54321	5
0			

Mô tả thuật toán



Cài đặt thuật toán

Dùng Do While ... Loop	Dùng Do ... Loop Until ...
t = 0 r = 0 Do While n > 0 r = n mod 10 t = t * 10 + r n = n / 10 Loop Print t	t = 0 r = 0 Do r = n mod 10 t = t * 10 + r n = n / 10 Loop Until n = 0 Print t

Thiết kế chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới.



Mã chương trình như bên dưới

```
Private Sub cmdDaoNguoc_Click()
    'Khai bao bien
    Dim n As Long
    Dim t As Long
    Dim r As Long

    'Lay du lieu
    n = Val(txtN.Text)

    'Xu ly
    t = 0
    r = 0
    Do While n > 0
        r = n Mod 10
        t = t * 10 + r
        n = n / 10
    Loop

    'Xuat ket qua
    lblKetQua.Caption = t
End Sub
```

Bài 6.7: Cho số nguyên n , cho biết n có phải là số nguyên tố hay không?

Phân tích

Số nguyên tố là số nguyên lớn hơn 1 chỉ có hai ước là 1 và chính nó.

Để kiểm tra điều này ta chỉ cần lấy n chia cho các số 2, 3, ..., $n-1$. Ngay khi ta bắt gặp một số nào là ước của n ta có thể dừng và xác định n không phải là số nguyên tố. Ngược lại, nếu không có số nào trong các số 2, 3, ..., $n-1$ là ước của n thì n là số nguyên tố. Cách làm này dựa vào định nghĩa trên của số nguyên tố. Tuy nhiên, liệu ta có thể đề xuất một thuật toán hiệu quả hơn?

Giả sử nếu n không phải là số nguyên tố, nghĩa là n có thể được phân tích dưới dạng $n = a \times b$, với $a > 1$ và $b > 1$. Khi đó, $a \leq \sqrt{n}$ hoặc $b \leq \sqrt{n}$ (vì nếu không thì $a \times b > n$). Như vậy, nếu n không phải là số nguyên tố thì n có ít nhất một ước $\leq \sqrt{n}$. Điều này cũng có nghĩa là nếu n không có ước nào $\leq \sqrt{n}$ thì n là số nguyên tố.

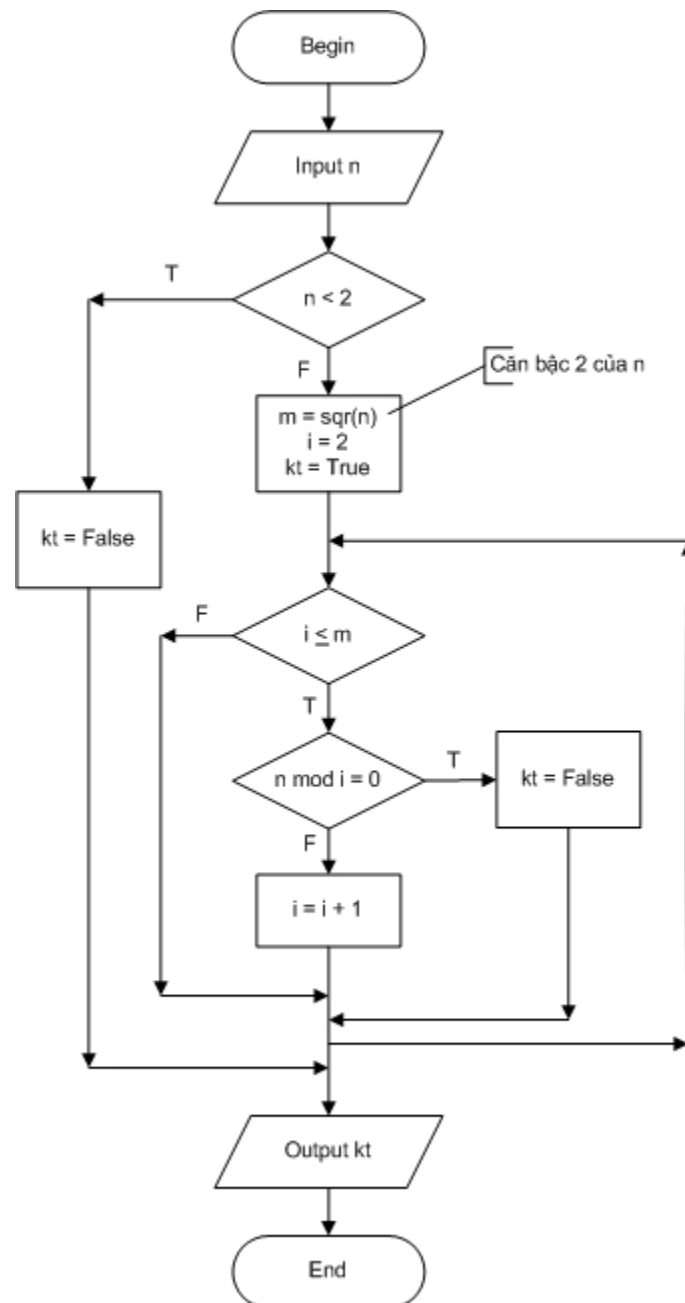
Như vậy, để kiểm tra xem n có phải là số nguyên tố hay không, ta chỉ cần lấy n chia cho các số $2, 3, \dots, \sqrt{n}$ ($\sqrt{n}$ nhỏ hơn đáng kể so với n khi n lớn). Ngay khi ta bắt gặp một số nào là ước của n ta có thể dừng và xác định n không phải là số nguyên tố. Ngược lại, nếu không có số nào trong các số $2, 3, \dots, \sqrt{n}$ là ước của n thì n là số nguyên tố.

Mô tả thuật toán

Xem lưu đồ bên dưới

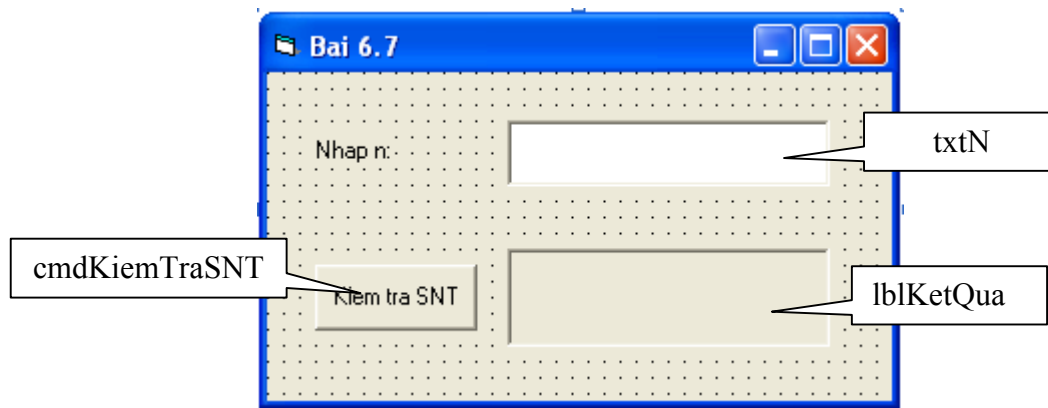
Cài đặt thuật toán

Dùng For	Dùng Do While ... Loop	Dùng Do ... Loop Until ...
If $n < 2$ Then $kt = \text{False}$ Else $m = \text{sqr}(n)$ $kt = \text{True}$ For $i = 2$ To n If $n \bmod i = 0$ Then $kt = \text{False}$ Exit For End If Next i End If Print kt	If $n < 2$ Then $kt = \text{False}$ Else $m = \text{sqr}(n)$ $i = 2$ $kt = \text{True}$ Do While $i \leq m$ If $n \bmod i = 0$ Then $kt = \text{False}$ Exit Do End If $i = i + 1$ Loop End If Print kt	If $n < 2$ Then $kt = \text{False}$ Else $m = \text{sqr}(n)$ $i = 2$ $kt = \text{True}$ Do If $n \bmod i = 0$ Then $kt = \text{False}$ Exit For End If $i = i + 1$ Loop Until $i > m$ End If Print kt



Thiết kết chương trình hoàn chỉnh

Thiết kế giao diện chương trình và đặt tên các đối tượng như hình bên dưới.



Mã chương trình như bên dưới

```
Private Sub cmdKiemTraSNT_Click()
    'Khai bao bien
    Dim n As Integer
    Dim i As Integer
    Dim m As Integer
    Dim kt As Boolean

    'Lay du lieu
    n = Val(txtN.Text)

    'Xu ly
    If n < 2 Then
        kt = False
    Else
        m = Sqr(n)
        kt = True
        For i = 2 To m
            If n Mod i = 0 Then
                kt = False
                Exit For
            End If
        Next i
    End If

    'Xuat ket qua
    lblKetQua.Caption = kt
End Sub
```