Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Fundamentals of Computer Science
2010-2011
Laboratory 7
Loops: Revision

erran ta zabal zazu

Objectives:
- To go deeply into the **design** of flowcharts with **repetitive structures**
- To go deeply into the **program coding** with **repetitive structures**

Program to demonstrate the use of repetitive sentences

Interface

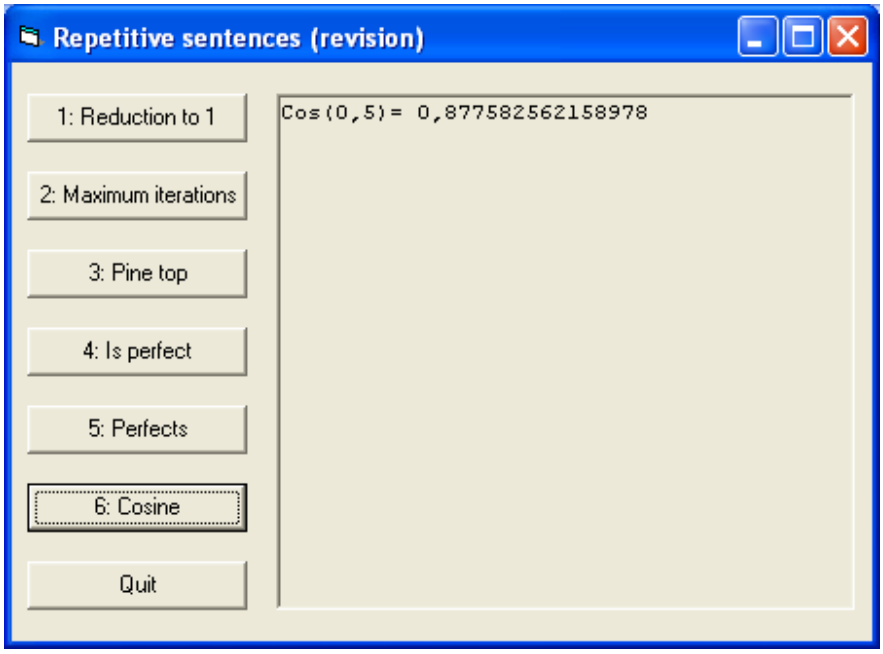


Figure 7.1 Objects present in the interface: command buttons and picture box

Operation

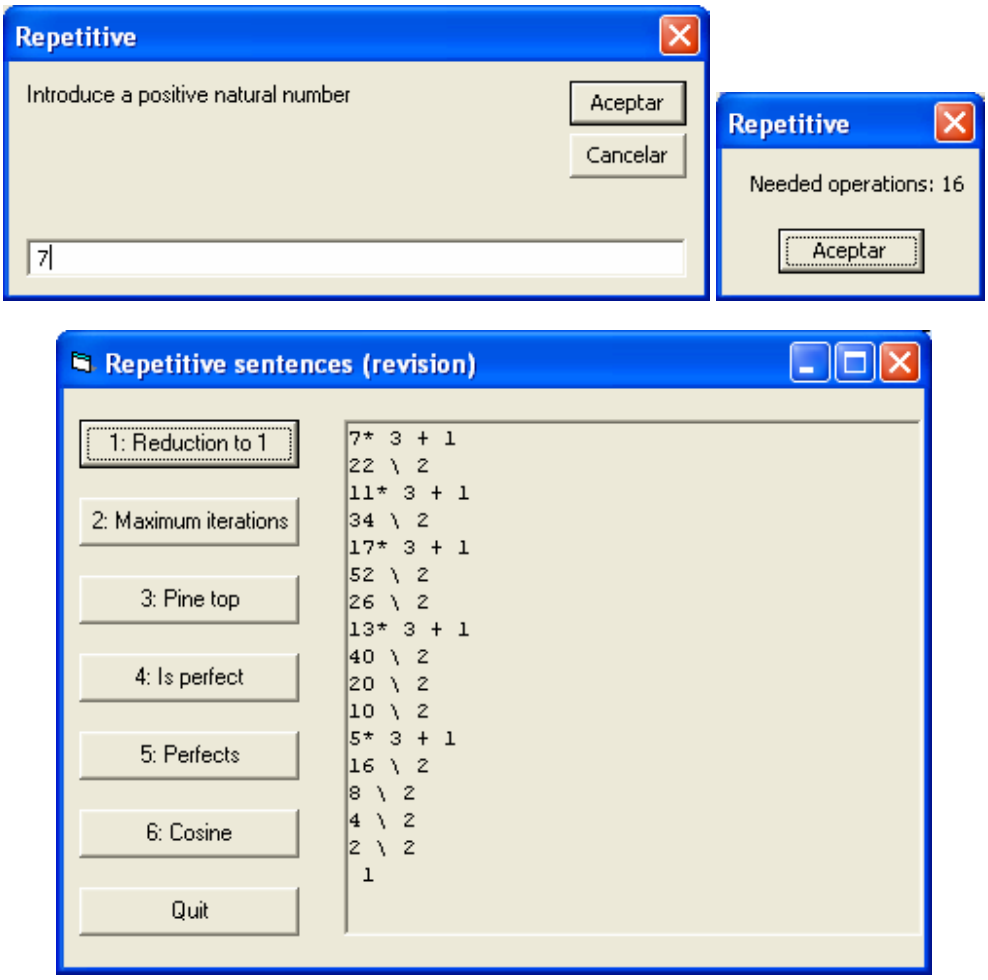
1. Each exercise has its own execution button (**`cmdEx1`**, **`cmdEx2`**, ..., **`cmdEx6`**).
2. To display the result we use a picture box, **`pctRes`**. For a better appreciation of the result the default font for the picture box must be "**`Courier New`**" size **8**.
3. First thing after clicking one button will be removing the contents of the results picture box, **`pctRes`**. To do so we use the **`Cls`** method (`pctRes.Cls`).
4. When we click on the Quit button the program will finish.
5. An executable is provided to clarify the statements.

Exercise 7.1: Reduction to one (partially solved)

Enunciation

Starting from any positive number **n** and applying successively the **division by two** (n **Div** 2) when the number is **even** and the **multiplication by three and addition of 1** ($n \cdot 3 + 1$) when the number is **odd** we reach number 1.

Design the **flowchart** and implement a program to read **n** and carry out these computations displaying at the picture box the operations carried out and showing at the end the number of iterations needed through a **`MsgBox`**. Note that after a given number of iterations there will be no space left at the picture box for more text.



Figur3 7.2 Example of reduction to 1 of number 7

Flowchart

For this exercise we propose to obtain the flowchart after the VB code.

Visual Basic code (resolution)

```
Sub cmdEj1_Click()
  Dim s As String
  Dim n As Integer
  Dim i As Integer
  pctRes.Cls
  s = InputBox("Introduce a positive natural number")
  n = CInt (s)
  i = 0
  While n <> 1
    i = i + 1
    If n Mod 2 = 0 Then
      pctRes.Print CStr (n) & " \ 2"
      n = n \ 2
    Else
      pctRes.Print CStr (n) & "* 3 + 1"
      n = n * 3 + 1
    End If
  Wend
  pctRes.Print CStr (n)
  MsgBox " Needed operations: " & CStr (i)
End Sub
```

Figure 7.3 VB code for the reduction to 1

1. **Design** the flowchart for the reduction to 1 program, partially solved.

2. **Design** the flowchart and **implement** the VB program to read a **top** number and verify which of the numbers from 1 to **top** require the **maximum** amount of **iterations** to reduce to 1 as in exercise 7.1.

3. **Design** the flowchart and **implement** the VB program to read a positive number **n** and draw the top of a pine tree of dimension **n**. Figure 7.4 shows some examples. Note: to prevent the new line to the Print order on a picture box you can add a semicolon (;) at the end of the text to write, e.g. `ptc1.Print "*";`

*	* ***	* *** *****	* *** ***** *******
n = 1	n = 2	n = 3	n = 4

Figure 7.4 Examples of pine tops of dimensions 1, 2, 3 and 4.

4. **Design** the flowchart and **implement** the VB program to read a positive number **n** and say if that number is **perfect**. A number **n** is called perfect when the sum of its divisors (except **n** itself) equals that number **n**. For example, 6 is perfect because 1+2+3 = 6.

5. **Design** the flowchart and **implement** the VB program to read a **top** number and verify which numbers from 1 to **top** are **perfect**.

6. **Design** the flowchart and **implement** the VB program to read an **angle** in radians and calculate its **cosine** using Taylor series with an error (absolute value of the difference between two approximations) less than 0.000001.

$$\cos(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}$$

Testing data

Angle (radians)	Cosine
1,0	0,540302303791887
1,5	7,07372049851851**E-02**
2,0	-0,416146839638903

Algorithm cosine (resolution)

Figure 7.5 shows the flowchart of a possible resolution to the problem.

The type of variable **fact2i** is **real** to allow a better resolution because for a value of i=7 an overflow occurs with integer values.

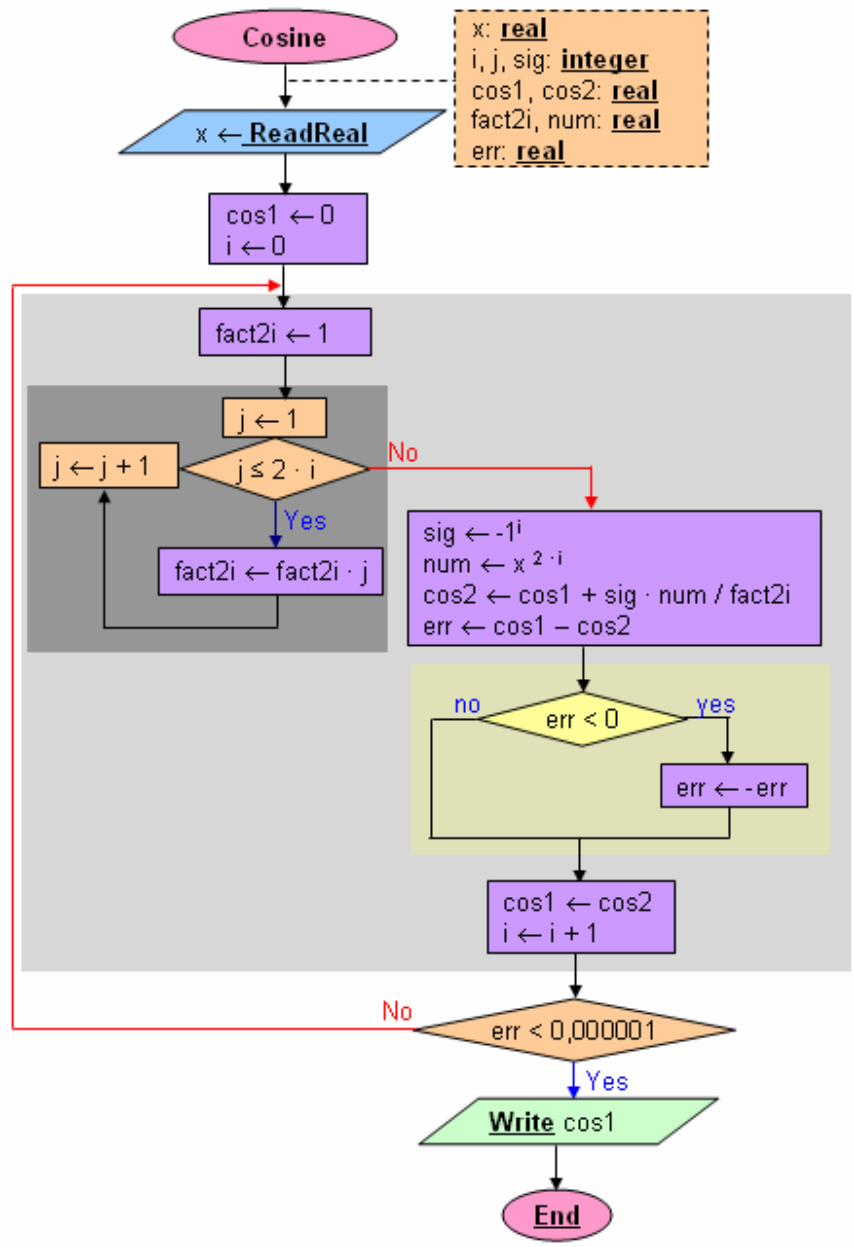


Figure 7.5 Flowchart of the cosine calculation.