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

Fundamentals of Computer Science
2010-2011
Laboratory 5
Loops: For sentences

erran ta zabal zazu

Objectives:
- **Design** flowcharts and **implement** VB programs with **repetitive structures**
- Design and implement VB programs with **`For` loops**
- Design and implement VB programs combining **conditional** and **repetitive** structures
- **`Long`** type

Program to demonstrate the use of For sentences

Interface

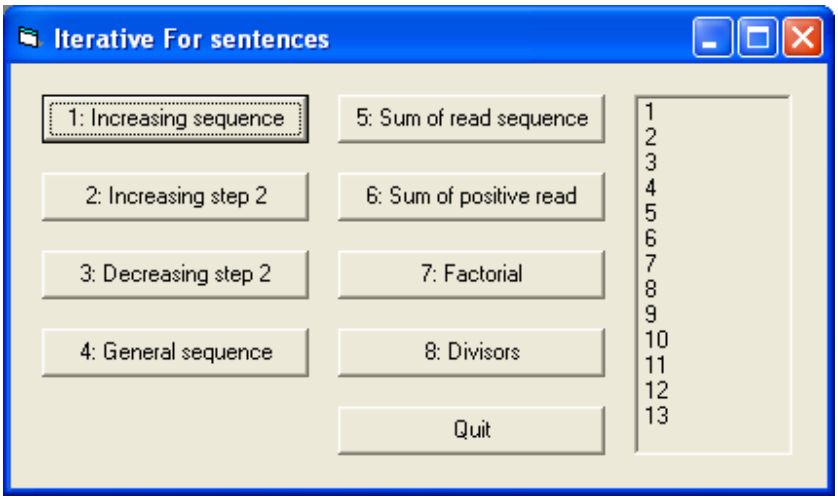


Figure 5.1 Objects present in the interface: command buttons and picture box

Operation

1. Each exercise has its own execution button (**`cmdEx1`**, **`cmdEx2`**, ..., **`cmdEx8`**).
2. The first action on the click event on one button will be to remove the contents of the results picture box, **`pctRes`**. To do so we use the **`Cls`** method (`pctRes.Cls`).
3. When we click on the Quit button the program will finish.
4. An executable is provided to clarify the statements.

Exercise 5.1: write an incremental sequence of numbers (resolution)

Operation

1. When the user clicks on the button labelled "1: Increasing sequence" the program will ask for the upper limit and then print a sequence of numbers from 1 to the maximum provided, one by one, each in a different line as shown in figure 5.1 for a limit of 13.

Flowchart

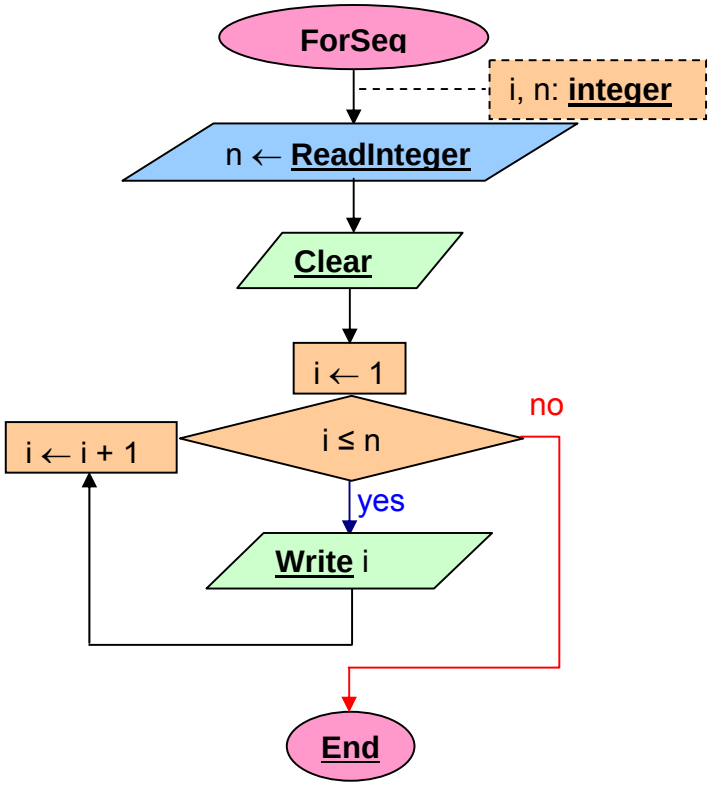


Figure 5.2 Flowchart of the first exercise using a `For` statement.

Steps

1. We create the objects as in Figure 5.1.
2. Add the code associated to the events. The code associated to the first exercise is shown in Figure 5.3.

```
Sub cmdEj1_Click()
  Dim s As String
  Dim i As Integer, n As Integer
  s = InputBox("Number of elements")
  n = CInt (s)
  pctRes.Cls
  For i = 1 To n Step 1
    pctRes.Print i
  Next i
End Sub
```

Figure 5.3 VB code for the incremental sequence with step 1.

Proposed exercises

Having solved the first exercise we propose the resolution for the rest. You don't need to verify if numbers are numeric if you are not asked to do it.

2. **Design** the flowchart and **implement** the VB program to read a **max** number and then display the increasing sequence from two to the given number with a step of two (two by two) in the picture box.

3. **Design** the flowchart and **implement** the VB program to read a number and then display the decreasing sequence from the given number to 1 with a step of two (two by two) on the picture box. **Verify** that the datum read is **numeric** and **positive**.

4. **Implement** the VB program to read the initial number **ini**, a final number **fin** and a step **stp**, **verifying that they are numeric**, and then display the sequence from **ini** to **fin** with a step of **stp** on the picture box. If **stp** is null it will show an error message. With no extra programming (the normal functioning of a `For` loop), if **stp** is positive there will be an increasing sequence and if it is negative it will be decreasing. If there is not any number between **ini** and **fin** it will show no number. **Check the provided demo program** to show the same error messages under the same circumstances. This behaviour cannot be represented in a generic way on the flowchart, as the stopping condition should have the ≤ on one case and ≥ on the other.

5. **Design** the flowchart and **implement** the VB program to read the number elements to add **n** and ask them ($\mathbf{e_1}$, $\mathbf{e_2}$, …, $\mathbf{e_n}$), showing the final sum $\mathbf{e_1 + e_2 + e_3 + ... + e_n}$. The elements will be of type `Double`.

6. **Design** the flowchart and **implement** the VB program to read the number elements to add **n** and ask them, showing the final **sum** $\mathbf{e_1 + e_2 + e_3 + ... + e_n}$ of the **positive** ones **ignoring negatives**. The elements will be of type `Double`.

7. **Design** the flowchart and **implement** the VB program to ask for a natural number **n** and calculate its factorial **n!**. Check the following cases: 0! = 1, 1! = 1, 4! = 24, 9! = 362880. It is recommended to use long integers (`Long`). Note that after the factorial of 13 there is an overflow.

8. **Design** the flowchart and **implement** the VB program to ask for a natural number **n** and show its **divisors** starting from two.

Quick reference table

Syntax	Example	Flowchart

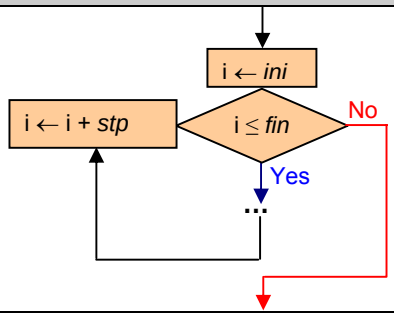
`For` i= *ini* `To` *fin* `Step` *stp*
  ...
`Next` i
	`For` i=1 `To` 5 `Step` 1
	  pctRes.`Print` i
	`Next` i
		

Table 5.1 Syntax for iterative structures type `For`.