

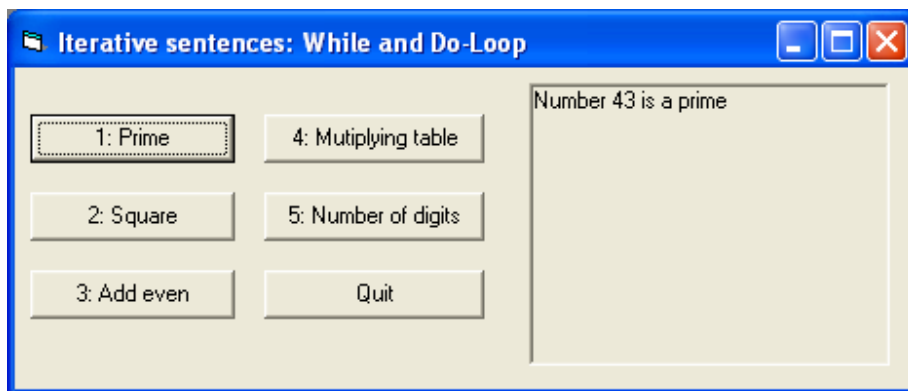


**Objectives:**

- ❖ Design the flowchart of programs with While y Do-Loop
- ❖ **Implement** VB programs with While y Do-Loop
- ❖ Combine conditional and repetitive sentences

## Program to demonstrate the use of While and Do-Loop sentences

### Interface



**Figure 6.1** Objects present in the interface: command buttons and picture box

### Operation

1. Each exercise has its own execution button (`cmdEx1`, `cmdEx2`, ..., `cmdEx5`).
2. 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`).
3. When we click on the `Quit` button the program will finish.
4. An executable is provided to clarify the statements.

## Exercise 6.1: given a positive number say if it is a prime<sup>1</sup>

### Operation

When the user clicks on the button “1: Prime” the program asks for a positive number and it will display in the picture box if that number is a prime or not, as shown in Figure 6.1.

<sup>1</sup> Natural number with two different natural divisors: 1 and itself. By definition 1 is not a prime.

## Algorithm

There are several algorithms to say if a given number  $n$  is a prime. Here we propose one consisting in finding the first number  $d$  that starting from 2 is divisor of  $n$ . If  $d$  reaches  $n$  it means that the number is a prime. This algorithm is valid for numbers greater than 1. Number 1 is by definition not a prime. Figure 6.2 shows the flowchart for a program to read and check if a number greater than 1 is a prime.

## Flowchart

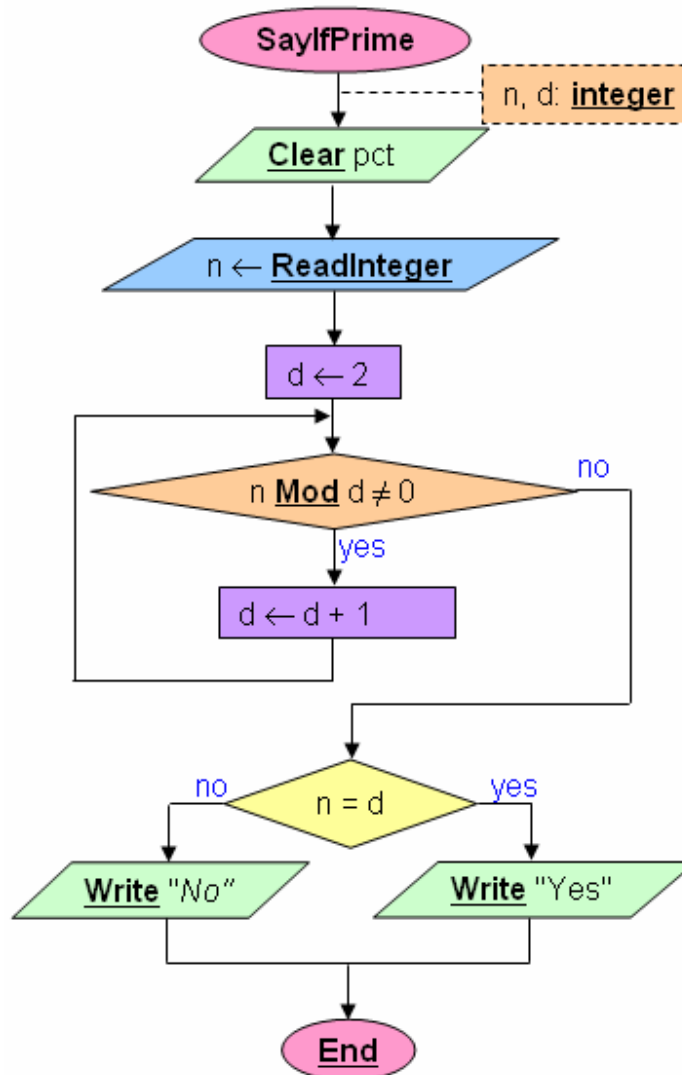


Figure 6.2 Flowchart for exercise 1

## Steps

1. We create the objects as in Figure 6.1.
2. Add the code associated to the events. The code associated to the first exercise is shown in Figure 6.3.

```
Sub cmdEj1_Click()  
Dim n As Integer, d As Integer  
pctRes.Cls  
n = InputBox("Introduce a positive number")  
d = 2  
While n Mod d <> 0  
    d = d + 1  
Wend  
If n = d Then  
    pctRes.Print "Number " & n & " is a prime"  
Else  
    pctRes.Print "Number " & n & " is not a prime"  
End If  
End Sub
```

Figure 6.3 VB code to say if a number greater than 1 is a prime

### Proposed exercises

1. **Implement** the VB program (mostly solved) of the **prime numbers** verifying that the number introduced is **numeric** and **positive**. Remember that 1 is not a prime so make a **special case**. For other non-prime numbers specify which is the first divisor found.
2. **Design** the flowchart and **implement** the VB program to read a sequence of numbers ended by 0 and show at the picture box the square of each number except the finishing 0.
3. **Design** the flowchart and **implement** the VB program to read a sequence of natural numbers ended by 0 and show at the picture box the **sum** of the **even** numbers.
4. **Design** the flowchart and **implement** the VB program to ask for and read a number **n** and show at the picture box each time the **multiplication table** from 1 to 9 of that number. The program will finish when a null value is read.
5. **Design** the flowchart and **implement** the VB program to read a natural number and calculate the number of digits in base 10 of that number. Although the natural solution is to use Integer variables the use of Double variables is suggested to allow bigger quantities.

### Quick reference table

Syntax	Example	Flowchart
<p><u>While</u> <i>cond</i>                      ...  <u>Wend</u></p>	<p><u>While</u> <math>p &lt; n</math>  <math>p = p * 10</math>  <u>Wend</u></p>	
<p><u>Do</u>                      ...  <u>Loop While</u> <i>cond</i></p>	<p><u>Do</u>  <math>s = s + 1</math>  <math>n = n / 10</math>  <u>Loop While</u> <math>n &gt;= 10</math></p>	
<p><u>Do</u>                      ...  <u>Loop Until</u> <i>cond</i></p>	<p><u>Do</u>  <math>s = s + 1</math>  <math>n = n / 10</math>  <u>Loop Until</u> <math>n &lt; 10</math></p>	
<p><u>For</u> <math>i = ini</math> <u>To</u> <math>fn</math> <u>Step</u> <math>stp</math>                      ...  <u>Next</u> <math>i</math></p>	<p><u>For</u> <math>i = 1</math> <u>To</u> <math>5</math> <u>Step</u> <math>1</math>  <math>pctRes.Print\ i</math>  <u>Next</u> <math>i</math></p>	