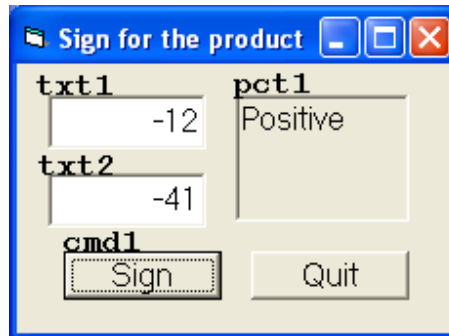


- (1 point) Not significant for the 2008-2009 academic year.
- (1.5 points) **Implement** a VB program to read two integer numbers (verifying if they are numeric) from two text boxes and write without carrying out the operation if their product results “Positive”, “Null” or “Negative”. Use for the controls the names proposed in the following figure.



- (3 points) **Implement** a VB function called CheckPass which receives a string containing a password and verifies, in the following order, if a series of conditions are fulfilled:
 - Length is between 6 and 10 characters (both included)
 - It has at least one digit (“0” – “9”)
 - It has at least a letter (“a” – “z”, “A” – “Z” with no tildes or special characters like ñ.
 - It has at least one character that is neither a digit nor a letter.

The function will return a code (only one) depending on whether it’s correct or one of the conditions is not fulfilled:

Code	Description of the code
0	Password correct
1	The length of the string is not between 6 and 10 characters
2	There is no digit
3	There is no letter
4	There is no character that is neither a digit nor a letter

- (3.5 points) **Design** the flowchart and **implement** a VB program to **calculate the value of π** (pi) with a given **precision ϵ** (epsilon) applying the “first lower term” method, described below.

The program will ask the user for the value of ϵ in between 0 and 1 (both excluded). You can suppose that the value introduced is numeric. If the value input is not in the valid interval it will show an error message and ask for this information again until we get a correct ϵ . At the end of the calculation it will show the result on the screen.

The π number may be calculated by means of the following summation of infinity terms:

$$\pi = 4 \cdot \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} \dots \right)$$

Note for the algorithm that the terms in even position are added and the terms in odd position are subtracted. The last term to add is the first one lower than ϵ .

Example:

If $\epsilon = 0.1$ we must calculate the sum $\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11}$ because $\frac{1}{11}$ is the first term lower than 0.1.

This sum results 0.744012; therefore the corresponding value of π is $\pi = 4 \cdot 0.744012 = 2.976046$.

The lower ϵ is the better approximation will be obtained. This way, if $\epsilon = 0.00001$ the corresponding value of π will be $\pi = 3.141416$.

5. (1 point) To know if there are substantial differences in the students marks we want to obtain the typical deviation σ (sigma). To do so we ask to **implement** a VB **function** that, receiving a *marks vector* \mathbf{v} , a previously calculated *arithmetic mean* \mathbf{m} and the *number of students* \mathbf{n} , calculates and returns a number, the typical deviation. The operation to be carried out by the function is:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - m)^2}$$

v_i refers to the given i^{th} mark of the vector

\mathbf{n} is the given number of students, although the vector may hold more marks

\mathbf{m} is the given arithmetic mean of marks in \mathbf{v}

Visual Basic reference table	
<code>IsNumeric(ByVal str As String) As Boolean</code>	Verifies if the string has a numeric value
<code>Val(ByVal str As String) As Double</code>	Numeric value of a string
<code>Mid(ByVal str As String, ByVal ini As Long, ByVal len As Long) As String</code>	Substring from <i>ini</i> until the given length <i>len</i> , or until the end of the string if not specified
<code>Len (ByVal str As String) As Integer</code>	String length
<code>Sqr (ByVal num As Double) As Double</code>	Square root of a number