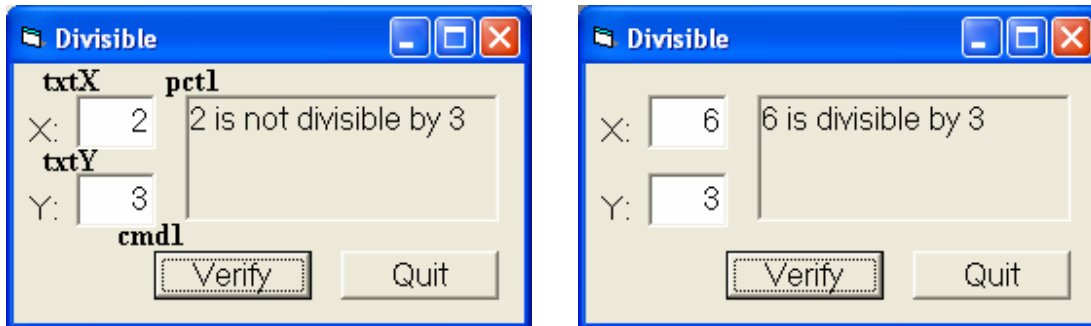


Name and Surname(s): _____

Previous notes:

- Write your **name and surname(s)** on this sheet and immediately on all supplementary sheets, even drafts. Not doing it may suppose your expulsion.
- You may use a **pencil** for your answers. You must turn your **mobile** off. You can neither use a **calculator**.
- All students implied in the copy of an exercise will have a final mark of 0. Students are responsible to take care of their own exams.
- “Receives” is different to “reads”. “Returns” or “obtains” is different to “writes”.
- Use comments in the declarations to indicate which variable corresponds to which concept.

1. (1.5 points) **Implement** the VB program given in the figure to read two positive natural numbers (verifying that they are **numeric** and **positive**) from two text boxes and write if **X is divisible by Y**. Use the proposed names for the controls in the figure (txtX, txtY, cmd1 and pct1).



Texts to write:

Text	Description	Example
The operands are not numeric	X or Y or both are not numeric	X: ddd
The numbers are not positive	X or Y or both are not positive	X: 0
6 is divisible by 3	X is divisible by Y	X: 6 Y: 3
2 is not divisible by 3	X is not divisible by Y	X: 2 Y: 3

2. (3 points) **Design** the **flowchart** for a program that asks the user for a **target number** and shows the first 5 numbers whose **sum of digits** gives the **target number**. To obtain the sum of the digits of a number use the method involving **integer divisions** (quotient/remainder). Examples:

Target number	Visualised message
12	39 48 57 66 75
20	299 389 398 479 488

Reference table 1 of Visual Basic functions	
<code>IsNumeric(ByVal str As String) As Boolean</code>	Verify if the string <i>str</i> contains a numeric value
<code>Val(ByVal str As String) As Double</code>	Numeric value of a string <i>str</i>

3. (3 points) MS Word text processing application provides, within its formatting options, the “Change Case” entry and, within this, the “Title Case” to convert the selected text to a title format. We want to automate a series of similar transformations to **normalise our titles**, correcting at the same time the duplication of separators and the use of these before punctuation signs.¹. Example:

Original string	Normalised string
“the repetitive sentences”	“The Repetitive Sentences”
“the HIG ant :that unknown creature”	“The Hig Ant: That Unknown Creature”

The main rules (verify with the provided examples) are:

- The original words are separated among them by separators and/or punctuation signs.
- All words in the normalised string are capitalised, that is, they start with a capital letter and the rest are in lowercase.
- A series of separators are substituted by a single space “ ”. Express it in your solution by means of an underscore “_” to make it different from an empty string.
- All separators before a punctuation sign are eliminated and after we will always add a space. Suppose that there may not be two punctuation signs together in the original string.

In addition to the standard VB function we also have the following `isSep` `isSig` functions:

<code>isSep (ByVal cha As String) As Boolean</code>	Given a character <code>cha</code> it says if it is a separator (space, tabulator, ...) or not
<code>isSig (ByVal cha As String) As Boolean</code>	Given a character <code>cha</code> it says if it is a punctuation sign (comma, dot, colon, ...) or not

You must:

- Implement** a VB function called `Titulise` that receives a string and returns a “normalised” version.
- Implement** a VB program (associated to a control button) that asks for a sentence and shows it normalised on the screen.

Reference table 2 of Visual Basic functions	
<code>Mid (ByVal str As String, ByVal ini As Long, ByVal len As Long) As String</code>	Substring from <code>ini</code> until the given length <code>len</code> , or until the end of the string if not specified
<code>Len (ByVal str As String) As Integer</code>	String length
<code>Ucase (ByVal str As String) As String</code>	Returns a copy of <code>str</code> in uppercase
<code>Lcase (ByVal str As String) As String</code>	Returns a copy of <code>str</code> in lowercase

4. (2.5 points) **Implement** a subprogram that, given two vectors of integer numbers sorted in ascending order and their sizes, obtains a third vector, equally sorted, which is a merge of the previous two. It also obtains its size. Example:

- `n1: 5, v1 = {-1, 4, 4, 7, 14}`
- `n2: 2, v2 = {7, 21}`
- **Result:** `n3= 7, v3 = {-1, 4, 4, 7, 7, 14, 21}`

¹ The result is not necessarily “correct”