**Task 1 – SuperCalculator**

Your task is to write a C# program to calculate the sum of two numbers provided by the user in words (not by digits).

The form should have two textboxes to get the numbers and a button to add them up. The input can only be English one-digit numbers written in lower case (“zero”, “one”, “two”, “three”… up to “nine”). Your program should calculate the sum of those numbers and display the result via Label control. User input should be validated, meaning input other than one-digit numbers is not valid. Corresponding error message should be displayed. **40 marks**

**Task 2 – Numerology**

Many people believe in numerology – the way to interpret various numbers associated with person’s life like birthdate, name, address etc. Your task is to write a C# program to automate one of such believes according to the following rules:

1. You should calculate the “destiny number” – sum of all components (day, month, year) of a person’s date of birth. For example if you were born on 01/02/2003 your “destiny number” is 0+1+0+2+2+0+0+3 = 8
2. If the “destiny number” is more than 10 you should add up the digits of that number. If result is again more than 10 you should repeat the process until the result is between 1 and 9. For example for 31/08/1998 => 3+1+0+8+1+9+9+8 = 39 => 3+9 = 12 => 1+2=3
3. It is well known that people with “destiny numbers” between 1 and 4 are “vampires”, 5 to 7 are “neutral” and with numbers of 8 and 9 are “angels”. Therefore a person born on 31/08/1998 is a “vampire” (see calculation above – “destiny number” is 3).

Your program should accept the date of birth via a textbox, compute “destiny number” and decide the type of person (“vampire”, “neutral” or “angel”) by that number. “Destiny number” and person type should be shown in corresponding label controls. User input should be validated, user-friendly error message should be displayed in case of wrong input.

## 60 marks

## 

|  |
| --- |
| **Assessment Criteria** |

|  |  |
| --- | --- |
| Component | Weight (marks) |
| Task 1 | 40 |
| Program has no syntactical errors | 5 |
| Comments present, variables are properly named | 5 |
| User input is validated, proper error messages are displayed | 10 |
| Calculation logic is done correctly. All operations are implemented | 15 |
| Output is done in proper way | 5 |
|  |  |
| Task 2 | 60 |
| Program has no syntactical errors | 5 |
| Comments present, variables are properly named | 5 |
| User input is validated, proper error messages are displayed | 20 |
| Calculation logic is done correctly | 20 |
| Output is done in proper way | 10 |