Tasks:

1) FizzBuzz. For each integer number n from 1 to 100 write the following: if n divisible by 3 write 'Fizz', if n divisible by 5 write 'Buzz', if n divisible by 15 write 'FizzBuzz', otherwise write its value.

2) Detect if string (latin alphanumeric, read it from console) starts with vowel (a, e, i, o, u, y) or not. Detect if it starts with digit.

3) Find n-th Fibonacci number. Read n from console, handle invalid (not numeric) input. Optional: handle oct and hex numbers (with prefix 0 and 0x accordingly).

4) Simple console calc (+ and -, only two integer numeric operands, handle invalid input, expression given in infix notation in one line, e.g. 1+3, 10- 42, 6 + 7).

5) Open txt file, count number of "words" (sequence of any characters split with any number of whitespaces). Better to do it without loading whole file to memory (use "floating window" approach). Generate input file automatically (easy, optional).

6) Count tab-index of xml or json file. Optional: cs file. (File tab-index is a maximum number of nested tags/scopes. E.g. <Node1/> has tab-index 0, <Node1><Node2/></Node1> has tab-index 1.)

7) Compare versions given in format x.y.z.t, e.g. 2.0.1.59, write which is greater. For instance 1.0 greater than 0.7.5.9, 2.4.6.0.7 less than 2.4.5.0.7, 1 equals to 1.0. Any number of version parts is valid. Delimiter is dot ('.').

8) Check if file has balanced braces, brackets, curly braces, triangle braces ('(' and ')', '[' and ']', '{' and '}', '<' and '>'). File may contain not only mentioned symbols. Example: "()[{}]<()>" is balanced, "{[]})<>()" or "<[{]}>" is not.