Lab 1a:

Statement: Considering a small programming language (that we shall call mini-langauge), write 3 small programs in this language.

Deliverables: p1.* , p2.*, and p3.* and p1err.* - small programs written in your programming language (p1, p2, p3 should be lexically correct; p1err should contain 2 types of lexical errors).

For example:

- **p1 and p2: compute de max/min of 3 numbers; verify if a number is prime, compute gcd of 2 numbers, compute the solutions for a 2nd order equation, aso**
- **p3: compute the sum of n numbers, computer the max/min of n numbers**

p1. Max of 3 nrs

read(nr1)

read(nr2)

read(nr3)

to_integer(nr1)

to_integer(nr2)

to_integer(nr3)

if(what_type(nr1) /= integer OR what_type(nr2) /= integer OR what_type(nr3) /= integer):

        write("all 3 numbers should be integers")

        end

if(nr1>=nr2 AND nr1>=nr3):

        write(to_string(nr1) + "is the max")

        end

elif(nr2>=nr1 AND nr2>=nr3):

        write(to_string(nr2) + "is the max")

        end

write(to_string(nr3) + "is the max")

p1err. Max of 3 nrs

read(nr1)

read(nr2)

read(3nr)

to_integer(nr1)

```
to_integer(nr2)

to_integer(3nr)

if(what_type(nr1) /= integer OR what_type(nr2) /= integer OR what_type(3nr) /= integer):

        write("all 3 numbers should be integers")

        end

if(nr1>=nr2 AND nr1>=3nr):

        write(to_string(nr1) + "is the max")

        end

elif(nr2>=nr1 AND nr2>=3nr):

        write(to_string"nr2 + "is the max")

        end

write(to_string(3nr) + "is the max")
```

p2. Verify if a number is prime

```
read(nr)

to_integer(nr)

if(what_type(nr) /= integer):

        write(to_string(nr) + "is not an integer")

        end

if(nr>1):

        DEF x, y

        x := integer

        y := integer

        y = to_integer(nr/2) + 1

        for x in from_to(2, y-1):

                if(nr%x == 0):

                        write(to_string(nr) + "is not prime")

                        end

        write(to_string(nr) + "is prime")

else:

        write(to_string(nr) + "is not prime")
```

p3. Max of n nrs

read(n)

to_integer(n)

if(what_type(n) /= integer):

      write(to_string(n) + "is not an integer")

      end

DEF x, y

x := integer

y := integer

y = 0

for x in from_to(1, n):

      read(val)

      to_integer(val)

      if(what_type(val) /= integer):

            write(to_string(val) + "is not an integer")

            end

      if(val>=y):

            y = val

write(to_string(y) + "is the max")