Laboratory 1

Tiutiu Natan-Gabriel

```
<<hello.hs>>=
<<printMessage>>
@

<<message>>=
"HelloWorld!"
@

<<printMessage>>=
print(<<message>>)
@
```

Euclidean algorithm

The purpose of this program is to give an implementation, and a correctness proof, of the Euclidean algorithm for computing the greatest common divisor of a list of positive integers chosen by the user.

But tthe goals of this text are more broader.

Algorithm description:

As long as b,which is the second elemnt is not equal to 0,we call recursively euclidean(b,a%b).

We can observe that always second element becomes the first element in the new recursive call.If we would not inverse the position of the elements,we would get stuck. Ex.:a =1000,b=100 euclidean(1000,100)= euclidean(1000,100)= $\ldots =$ euclidean(1000,100)= $\ldots$.

The mathematical model for our function:

$$euclidean(a,b) = \begin{cases} euclidean(b, a\%b), if b > 0 \\ a, else \end{cases}$$

The proof of corectness is based on the following lemma:

If a=c(mod b) (1), then (a,b)=(c,b) (3)

Proof of this lemma:

From (1) we have b|a-c,so there is a y such that by=a-c.If there is a d such that d divides a and b ,then it will also divide c=a-by. =>any divisor of a and b is a divisor of c and b. (2) Suppose (a,b)=x and (c,b)=y.Using (2) we have that x|y and y|x ,so we have that (a,b)=(c,b).

If we put a=a%b(mod b) =>using lemma (3) we have that (a,b)=(a%b ,b)

```
<<euclidean>>=
euclidean a b =
    if (b>0)
        then (euclidean b (mod a b) )
        else a
@
```