mini language rules:

relational operators are: > , < , >= , <= , !=
arithmetic operators are: + , - , * , % , /

for creating a function you write the key-word fnct , - , the name of the function, followed by the list of parameters separated by ,
every line of code in that function needs to start by a tab
to call a function we write the function name and inside brackets the parameters

for returning a variable we write raspuns and than the variable name

instead of for we use iaCateUna, which iterates through a vector starting at position 0
instead of if we use daca
instead of else we use nuECazu

at the end of every comand we use : to separate the line of code

the function to print a result is sight, it accepts any parameters, no matter how many

p1).

```
fnct - nrMax : n1, n2, n3
    daca n1 > n2 :
        daca n1 > n3 :
            raspuns n1
        nuECazu :
            raspuns n3
    nuECazu :
        daca n2 > n3 :
            raspuns n2
        nuECazu :
            raspuns n3

sight(nrMax(1,2,3))
```

p2).

```
fnct - cmmdc : a, b
    panaCand a = b:
        daca a > b:
            a=a-b
        nuECazu:
            b=b-a

    raspuns a

sight(cmmdc(10, 20))
```

p3).

```
fnct - sumaSiMax : vector[]
    exista max=0, suma=0
    iaCateUna vector[] :
```

```
        suma=suma+vector[nr]
        daca max > vector[nr]:
            max = vector[nr]

    raspuns (suma, max)

sumaSiMax([1,2,3,4,5,6,7,8,9,10])

p1err).

fnct nrMinim : a, b
    daac a < b :
        raspuns a
    nuECazu :
        raspuns b
```