MODULO

Deze opdracht bestaat uit 2 delen.

1. We gaan eerst een getal splitsen in een veelvoud van een basisgetal en een rest.

```
getal = factor * basis + rest
```

```
voorbeeld met basis = 5 en getal = 19 : 19 = 3 * 5 + 4 , dus factor = 3 en rest = 4. We kiezen de factor zo groot mogelijk zodat altijd geldt: 0 <= rest < basis!
```

2. Daarna gaan we deze aanpak gebruiken om een getal te schrijven in een ander talstelsel.

Je krijgt bij deze opgave een header file met daarin de declaraties van de functies die je moet implementeren.

Codeer instructies:

Maak eerst de twee functies die **factor** en **rest** bepalen bij gegeven getal en basis. Bedenk goed welke operatoren hiervoor het meest geschikt zijn!

Bepaal / bedenk vervolgens een algoritme om een waarde te noteren in een gegeven talstelsel waarbij je de bovengenoemde functies gebruikt.

Voorbeeld de waarde 19 levert op '**34'** en **'201'** voor 5-tallig en 3-tallig stelsel respectievelijk.

```
Immers: 19_{10} = 34_5 en 19_{10} = 201_3
```

Gegeven is de volgende code :

<filenaam>.cpp:

```
#include <...>
#include "modulo.h"

... hier de functies uit de header file ...
int main()
{
    ...
    return 0;
}
```

MODULO

Condities:

We werken hier alleen met integer basisgetallen 2 t/m 9. (anders wordt de notatie een probleem !) Elk basisgetal buiten deze reeks levert "0" op.

Test je oplossing goed.

- Bedenk dat 0 in elk basis de string '0' oplevert.
- De notatie moet 'standaard' zijn.

Dat wil zeggen Most Significant Digit links en Least Sigificant Digit rechts!

Voorbeeld : $19_{10} = 201_3$

Merk op:

- 1. De inhoud van de 'main' routine is niet van belang voor de test.

 Je mag hier test-code opnemen voor de gemaakte functies, geheel naar eigen inzicht.
- 2. De 'signatuur' van de functies mag niet worden aangepast. Wijzigingen geven fouten in de test.

Tip:

Je kunt hier de **stringstream** klasse gebruiken om de correcte notatie op te bouwen. Bedenk zelf welke Includes <...> je dan moet toevoegen !!

Upload instructies:

Upload de volgende bestanden:

1. modulo.cpp