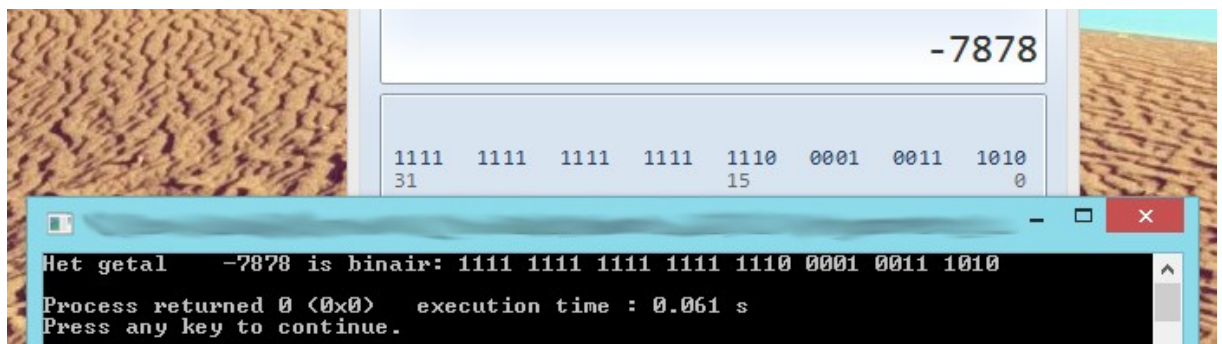


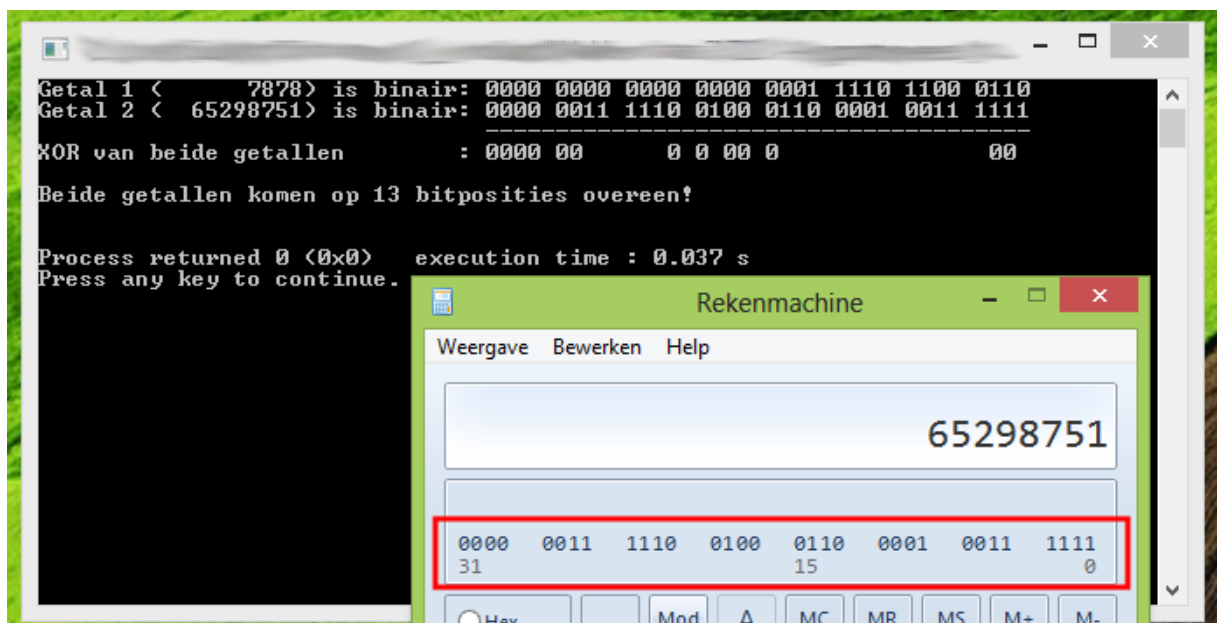
## Huiswerkopdrachten Programmeren in C - Week 2

De inleidende *én* uitleidende instructie bij de opdrachten van week 1 blijft ook voor deze week van kracht!

1. Schrijf en demonstreer een functie die een 32-bit integer omzet naar een 'binaire string' en die deze binaire representatie naar de standard output (= de console) kan printen. *De functie hoeft deze 'string' NIET als een returnvalue terug te geven, maar mag zelf printf aanroepen.* Bij voorkeur spatieer je de bitstring elke *nibble* voor de duidelijkheid. Het is **verplicht** om de functie op te nemen in een externe bibliotheek (met een eigen header file en .c definitie file). Deze library moet je vervolgens '#includen' in je demonstratiecode. Zie de screenshot voor een voorbeeld:



2. Schrijf en demonstreer een functie die van twee 32-bit integers bepaalt op hoeveel plaatsen (bits) beide integers gelijk zijn. Als ik bijvoorbeeld de integers 12 (1100) en 5 (0101) als argumenten meegeef, dan is de uitkomst van het onderzoek 30 (de middelste 2 bits komen overeen + alle 28 voorloophnullen). Gebruik de binary-print functie uit opdracht 1 om op de console een visuele controle weer te geven. Je kunt een hint voor de aanpak halen uit de screenshot hieronder.



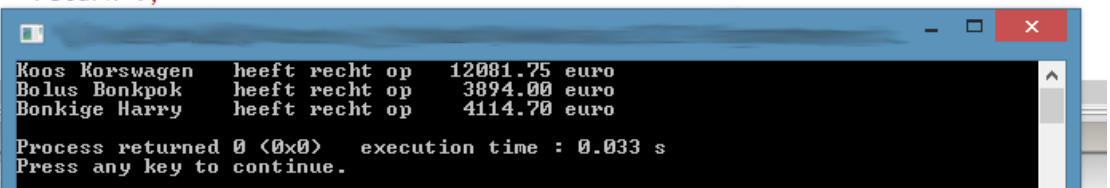
3. Schrijf en demonstreer een macro (m.b.v. de preprocessor directive '#define') die bepaalt of twee getallen aan elkaar gelijk zijn. In de demonstratiecode mag je ordinaire integers gebruiken voor de te

vergelijken getallen. Je bent echter verplicht om de macro een boolean-waarde terug te laten geven. Deze boolean moet je zelf d.m.v. een **enum** definiëren!

4. Maak een nieuw datatype aan dat een voornaam (max 50 karakters), achternaam (max 50 karakters), leeftijd (integer), uurtarief (double) en uren gewerkt (integer) bevat. Noem dit nieuwe datatype een 'opdrachtnemer'.

Schrijf en demonstreer een functie die van elk element uit een array van opdrachtnemers de beloning van elke opdrachtnemer berekent en toont. Uiteraard gebruik je een vorm van iteratie om door de array heen te lo(o)pen. Ook is het verplicht om 'puntnotatie' te gebruiken om de 'data-leden' van je datastructuur aan te spreken. De gewenste uitvoer en de 'main()' -functie van het programma zijn hieronder gegeven in een screenshot. *Zorg dat je de namen en bedragen net als in de screeny netjes uitlijnt door de 'printf()' -functie goed te gebruiken!*

```
26 int main()
27 {
28     opdrachtnemer opdrNemers[] = {
29         {"Koos" , "Korswagen", 32, 135.75, 89},
30         {"Bolus" , "Bonkpok" , 44, 162.25, 24},
31         {"Bonkige", "Harry" , 26, 89.45, 46}
32     };
33
34     int s = sizeof(opdrNemers)/sizeof(opdrachtnemer);
35
36     rapporteerBeloningen(opdrNemers, s);
37     return 0;
38 }
39
```



```
Koos Korswagen heeft recht op 12081.75 euro
Bolus Bonkpok heeft recht op 3894.00 euro
Bonkige Harry heeft recht op 4114.70 euro

Process returned 0 (0x0) execution time : 0.033 s
Press any key to continue.
```