## Opdracht 4.1 – Halve kerstboom

Schrijf een programma met een geneste for loop die volgend patroon van sterretjes op het scherm plaatst.

\*

\*\*

\*\*\*

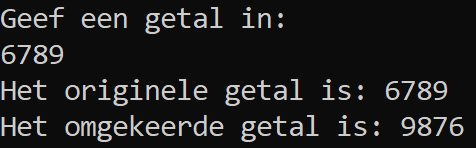
\*\*\*\*

\*\*\*\*\*

## Opdracht 4.2 – Omgekeerd Getal

We vragen aan de gebruiker om een geheel getal in te geven. Het programma zal het omgekeerde getal berekenen en dat op het scherm plaatsen.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **getal** | **kopieGetal>0** | **copyGetal** | **digit** | **copyGetal** | **omgekeerdGetal** |
| **1234** |  | 1234 |  | 1234 |  |
| **start while** |  |  |  |  |  |
| 1234 | true | 1234 | 1234 % 10=4 | 1234 / 10=123 | 0 \* 10 + 4 = 4 |
| 1234 | true | 123 | 123 % 10=3 | 123 / 10=12 | 4 \* 10 + 3 = 43 |
| 1234 | true | 12 | 12 % 10=2 | 12 / 10=1 | 43 \* 10 + 2 = 432 |
| 1234 | true | 1 | 1 % 10=1 | 1 / 10=**0** | 432 \* 10 + 1 = 4321 |
| 1234 | **false=STOP while** |  |  |  | 4321 |
| 1234 |  | 0 |  |  | **4321** |



## Opdracht 4.3 – Berekening intrest

Charlie Chaplin investeert 1000 EUR. In de gouden jaren kreeg hij nog 5% interest per jaar van de bank. We gaan ervan uit dat Charlie het geld 10 jaar onaangeroerd op de bank laat staan. Geef hierbij als output in welke jaar Charlie zit, en voor welk bedrag Charlie al op de bank heeft staan.

Gebruik onderstaande formule voor het berekenen van de intrest:

a = p (1+r)n

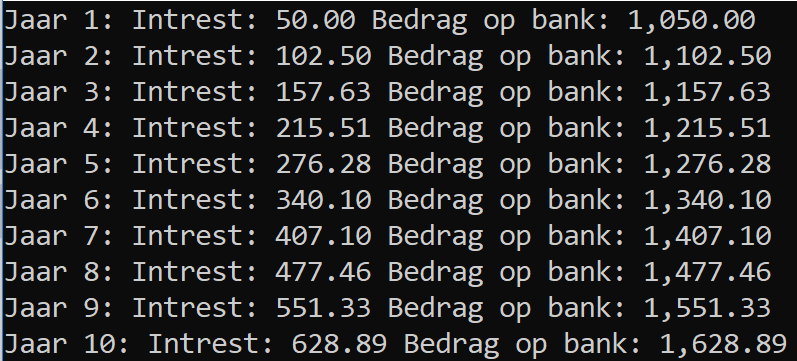
a: bedrag op de bank op het einde van jaar n

p: origineel geïnvesteerd bedrag

r: jaarlijkse intrest

n: aantal jaar

tip: bekijk hierbij eens de Math.Pow bewerking binnen .NET.

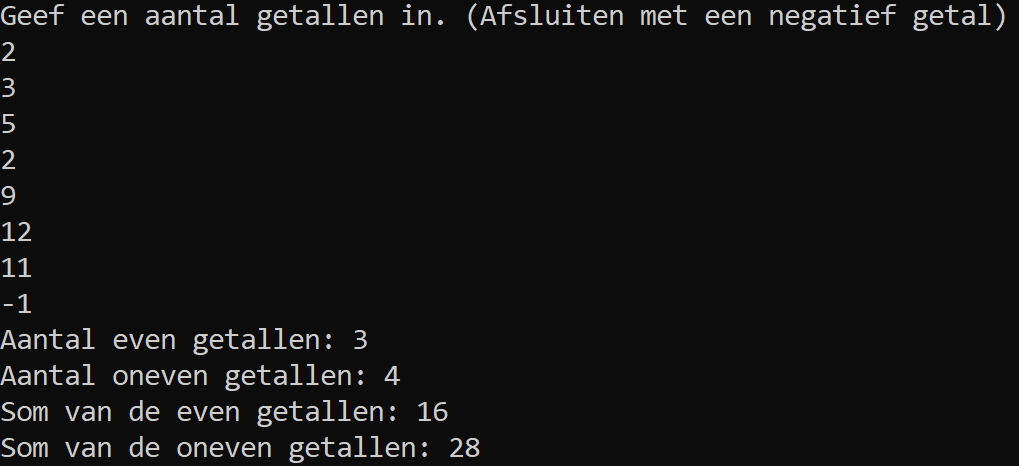


## Opdracht 4.4 – berekening resultaat even getallen

Andrei T. kan een oneindig aantal positieve getallen ingeven. De invoer stopt met het ingeven van een negatief getal.

Van de getallen die ingegeven worden, wordt weergegeven hoeveel even getallen er ingegeven werden alsook hoeveel oneven getallen. De som van de even en oneven getallen wordt eveneens weergegeven.

Gebruik hierbij een **do-while** lus.



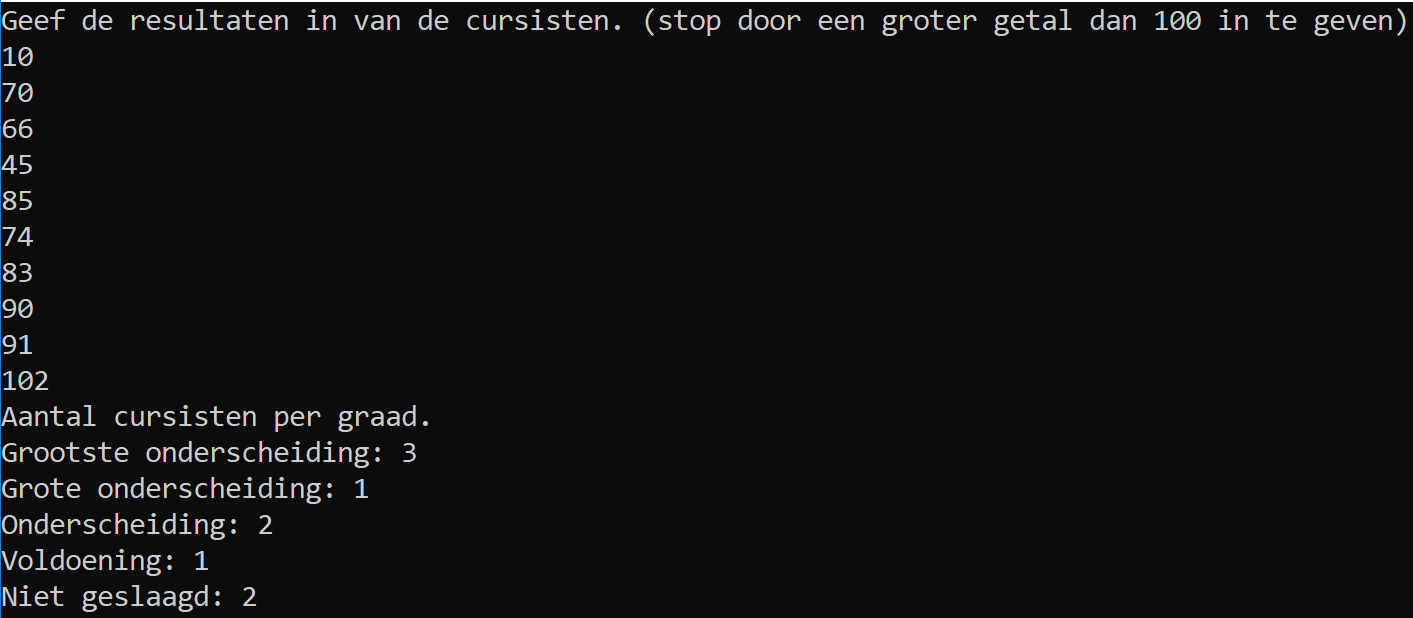
## Opdracht 4.5 – Omzetting cijfer resultaten naar codes

Een school wil een programma waarmee ze direct zien na het invoeren van de resultaten hoeveel cursisten er grootste onderscheiding, grote onderscheiding, onderscheiding, voldoening en niet geslaagd zijn.

Als input worden de resultaten ingegeven van de cursisten. De invoer wordt gestopt door een getal groter dan 100 in te geven.

Als output wordt het aantal resultaten die ingegeven zijn getoond, alsook het totaal van die resultaten. Ook het gemiddelde wordt getoond. En hoeveel cursisten er iedere graad hebben behaald.

* Grootste onderscheiding >=85%
* Grote onderscheiding >= 77%
* Onderscheiding >= 68%
* Voldoening >= 50%
* Niet geslaagd < 50%



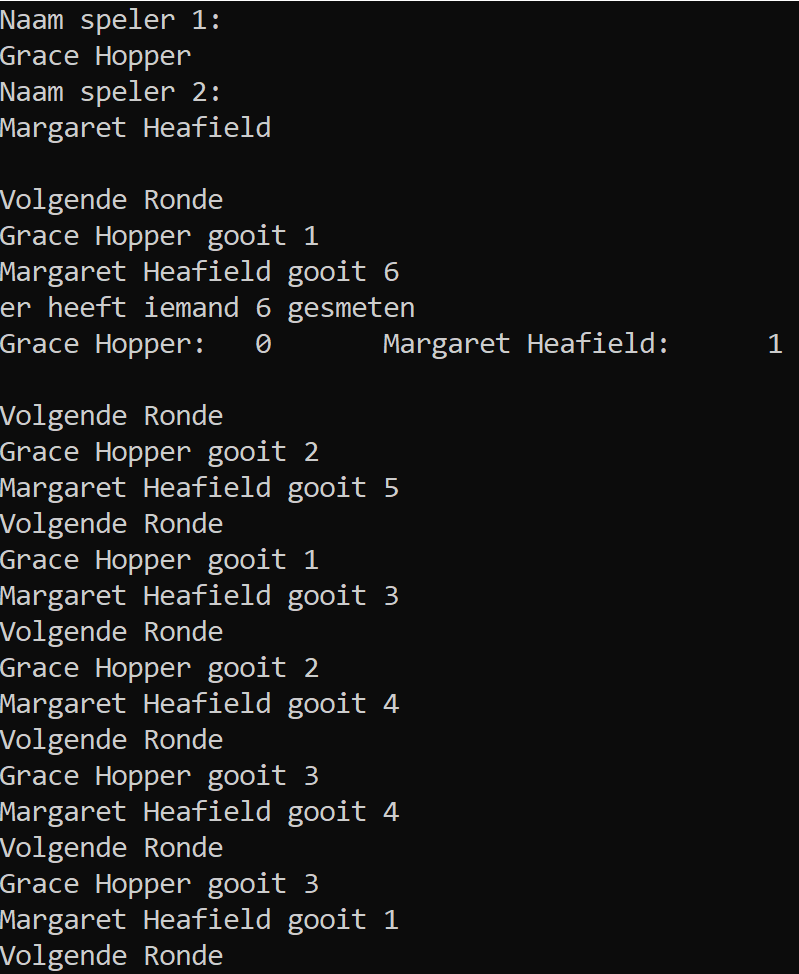
## Opdracht 4.6 - Dobbelsteen

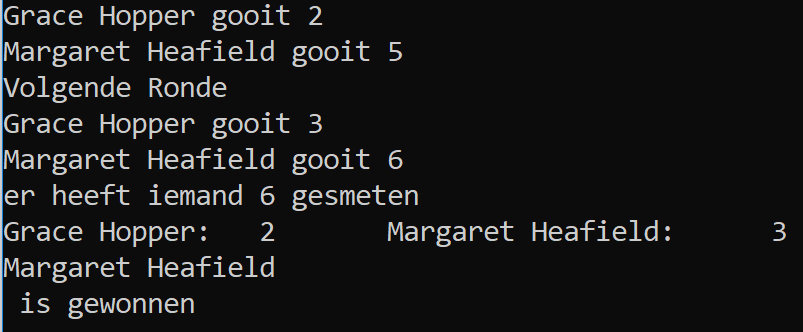
We schrijven een programma dat twee spelers laat spelen met een dobbelsteen. Het programma moet de volgende zaken kunnen doen:

1. De namen van twee gebruikers opvragen en bewaren
2. De gebruiker gooit de eerste dobbelsteen. Dus dat getal zal willekeurig door het programma worden berekend. (Zie oefening hoger lager voor voorbeeld om random getallen te laten genereren)
3. De waarde van de dobbelsteen die de eerste gebruiker heeft gegooid wordt bewaard in een variabele
4. De tweede gebruiker gooit de dobbelsteen. (maak hiervan weer gebruik van de methode van de willekeurige getallen). Dat getal zal weer door het systeem worden berekend.
5. De beide spelers kunnen blijven gooien tot er één van de twee of beide zes heeft gegooid. Wie zes heeft gegooid krijgt één punt.
6. Het spel is afgelopen als er één van de spelers 3 punten heeft behaald. (dus 3 maal de waarde 6 heeft gegooid).

In de uitvoer moet er telkens worden weergegeven wat de speler heeft gegooid. Indien hij een 6 heeft gegooid moet daar een melding van worden gegeven én de tussenscore moet worden getoond.

Bij afloop plaats je op het scherm wie heeft gewonnen.

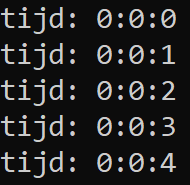
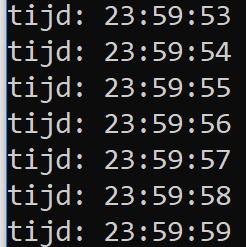
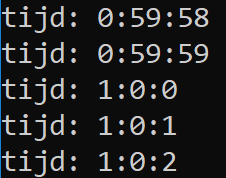
 



## Opdracht 4.7 – Uurwerk

Schrijf een programma dat een uurwerk simuleert. We starten de tijd met 00:00:00. Dit is in de vorm hh:mm:ss.

We gebruiken hiervoor drie for loops die genest zijn. Wanneer het een volledige 24 uur heeft gelopen hoef je niet meer te herhalen. Het programma zal dan stoppen.

## Opdracht 4.8 – Shopping

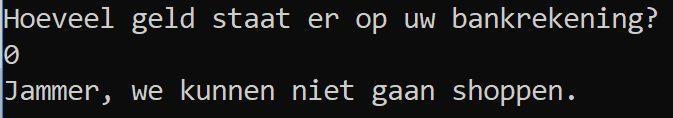
We vragen aan de klant hoeveel geld er op zijn bankrekening staat.

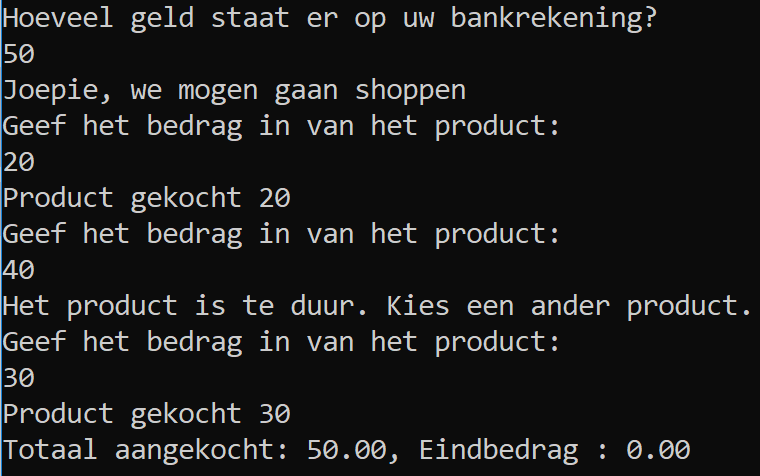
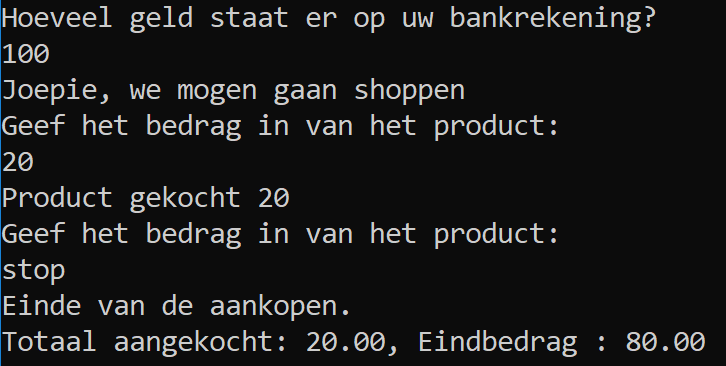
Als er geld opstaat kunnen we gaan shoppen.

De klant gaat de winkel binnen en legt een product in zijn winkelmandje. De verkoopster kan in het programma aangeven hoeveel het product kost. Zolang er voldoende geld op de bankrekening staat om het product te komen kan de klant het product kopen. Indien er geen genoeg geld meer beschikbaar is krijgt de gebruiker daarvan een melding.

Geef “Stop” in om te stoppen.

Op het einde krijgt de klant de totale prijs te zien die hij moet betalen én wat hij nog op zijn rekening over heeft.



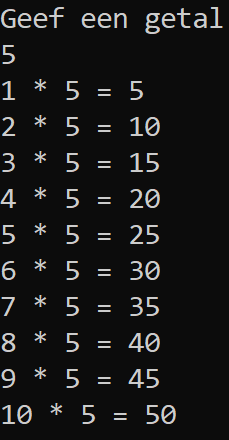
## Opdracht 4.9 – Ratten

In New York leven naar schatting 2 miljoen ratten. De populatie ratten groeit ieder jaar met 15 %. Ontwerp een consoletoepassing waarmee je kan berekenen hoelang het duurt vooraleer New York 10 miljoen ratten telt.



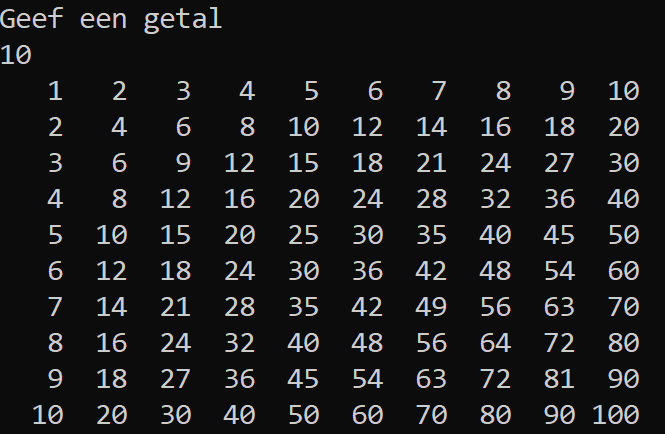
## Opdracht 4.10 – Tafels van vermenigvuldiging

Ontwerp een consoletoepassing waarmee de gebruiker de tafels van vermenigvuldiging verkrijgt van een ingegeven getal tussen 1 en 10.



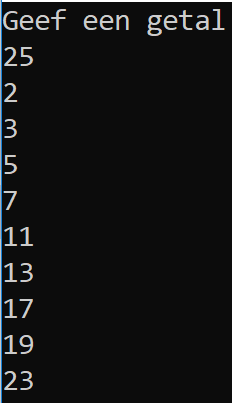
## Opdracht 4.11 – Vermenigvuldingsvierkant

Schrijf een programma dat een vermenigvuldigingsvierkant afdrukt. De grootte van het vierkant is afhankelijk van een in te lezen getal.



## Opdracht 4.12 – InvoerenGetallen

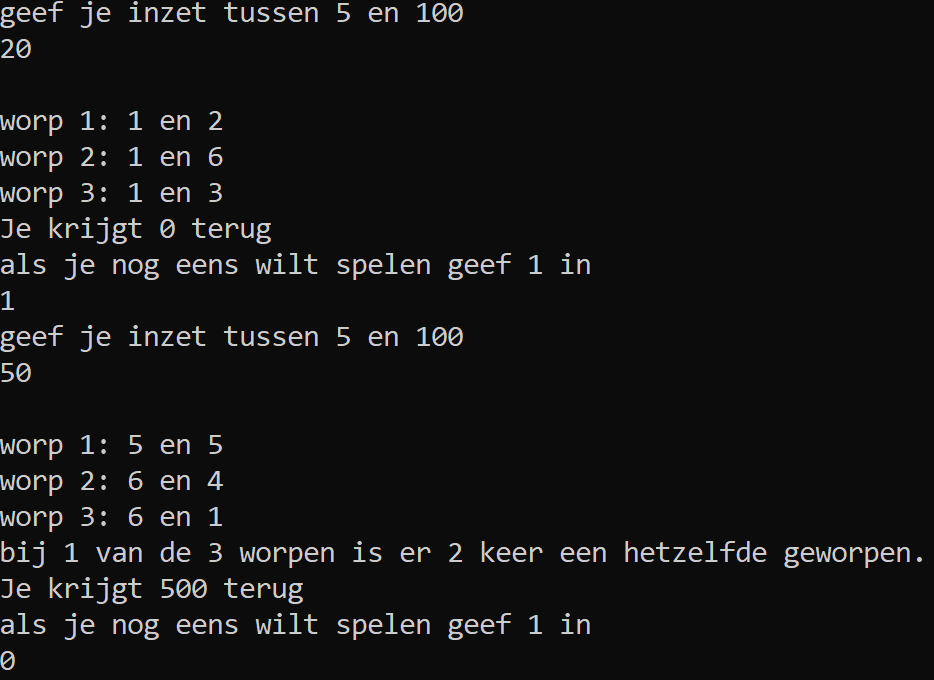
Ontwerp een consoletoepassing waarin de gebruiker een geheel getal kan invoeren met een maximum van 100. De console toont de getallen van 1 tot en met het ingevoerde.



## Opdracht 4.26 – Dobbelstenen

Ontwerp een consoletoepassing waarmee je 3 worpen met 2 dobbelstenen nabootst. Het resultaat in de console toont wat de deelnemer wint. De deelnemer kan zelf kiezen hoeveel hij inzet. De minimum inzet is 5 EUR en de maximum inzet is 100 EUR. Het spel kan herhaald worden na de laatste worp.

* bij 1 van de 3 worpen is er 2 keer een 6 geworpen: winst = inzet x 50
* bij 1 van de 3 worpen is er 2 keer hetzelfde geworpen: winst = inzet x 10
* bij de 3 worpen is er 2 keer een zes geworpen: winst = inzet x 2



## Opdracht 4.27 – Temperatuur

Het KMI ontvangt iedere dag temperatuurwaarnemingen uit het hele land. Deze gegevens worden per computer verwerkt. Er zijn niet elke dag evenveel waarnemingen.

Lees een willekeurig aantal temperaturen in (wanneer 99 wordt ingetikt, stopt de ingave). Druk hierna volgende resultaten af:

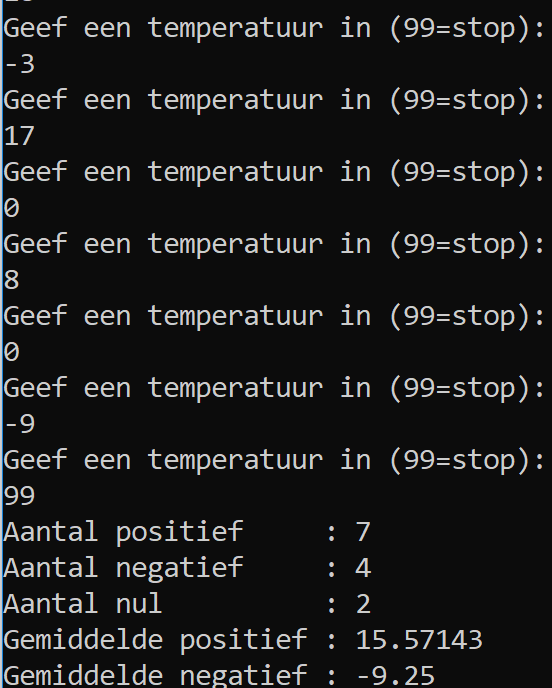
aantal positieve temperaturen,

aantal negatieve temperaturen,

gemiddelde van de positieve temperaturen,

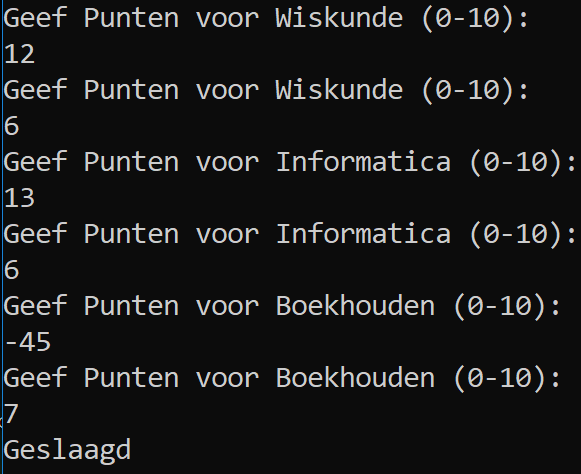
gemiddelde van de negatieve temperaturen,

aantal nul-temperaturen.

## Opdracht 4.28 – Examens

We hernemen opnieuw de opgave Examens (Zie opdracht 3.22). Lees de examenuitslagen in voor drie vakken (wiskunde, boekhouden en informatica). Elk van de vakken staat op 10 punten. Voeg de validatie toe bij het inlezen van de punten, zodat je punten kan blijven ingeven totdat je een juiste waarde hebt voor de punten van wiskunde, boekhouden en informatica. De student is geslaagd indien hij/zij voor wiskunde minstens 6/10 haalt, en voor boekhouden en informatica samen minstens 12/20. Toon op het scherm of de student geslaagd is, en indien de student niet geslaagd is, toon je ook de reden daarvoor.



## Opdracht 4.29 – Lidgeld

Een vereniging vraagt € 10,00 lidgeld per jaar. Leden ouder dan 50 jaar krijgen € 2,00 reductie. Per kind ten laste wordt € 1,00 reductie gegeven (met een maximum van € 5,00). Indien het jaarinkomen onder € 12 500,00 ligt, wordt € 2,50 korting gegeven. De maximale reductie per lid is € 8,50. Maak een programma dat de nodige gegevens inleest en het te betalen lidgeld toont. Het programma eindigt wanneer bij de naam ‘stop’ of ‘STOP’ ingegeven wordt.