

Beginnelen van Programmeren: Oefenzitting 2

iteraties: while – for

1. Schrijf een programma dat de som van de eerste n natuurlijke getallen afdrukt:
 $1 + 2 + 3 + \dots + n$.
2. Schrijf een programma dat een temperatuur in graden Celsius omzet naar graden Fahrenheit. Het programma moet invoer accepteren tot de gebruiker de letter 'q' intypt. Formule: $T[^\circ\text{F}] = 1.8 * T[^\circ\text{C}] + 32^\circ\text{C}$ (25°C komt dus overeen met 77°F .)
3. Schrijf een programma dat een piramide afdrukt. Vraag eerst de hoogte aan de gebruiker.
vb: hoogte = 4
 X
 XXX
 XXXXX
 XXXXXXX
4. Schrijf een programma dat een aantal getallen inleest (afgesloten door het getal 0) en de alternerende som berekent en afdrukt.
Voorbeeld: bij invoer van 1 4 9 16 9 7 4 9 11 0 wordt de som $1 - 4 + 9 - 16 + 9 - 7 + 4 - 9 + 11 = -2$ afgedrukt.
5. Schrijf een programma dat nagaat of een gegeven positief getal een priemgetal is.
(UOVT) Denkvraag: Kun je ook een efficiënter algoritme bedenken?
6. **(UOVT)** Pas het vorige programma aan zodat het alle priemgetallen kleiner dan het ingevoerde getal afdrukt.
7. Schrijf een programma dat de faculteit $n!$ van een gegeven getal n berekent.
 $n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot n$.
8. **(UOVT)** Schrijf een programma dat e^x berekent voor een gegeven getal x . Dit kan gevonden worden met de volgende formule:

$$e^x = \sum_{n=0}^{\infty} x^n / n!$$

Je hoeft geen oneindige som te berekenen, blijf gewoon termen toevoegen tot deze kleiner worden dan een bepaalde grenswaarde.

9. Schrijf een programma dat controleert of een ingegeven kredietkaartnummer geldig is. Kredietkaartnummers bestaan uit 16 cijfers, bijvoorbeeld: 1234 5678 1234 5670. Om te controleren of dit nummer effectief een kredietkaartnummer is wordt volgend algoritme toegepast:
 - 1) Label de cijfers van 0 t.e.m. 15.
 - 2) Verdubbel alle cijfers met even label. Telkens je op twee cijfers uitkomt tel je die bij elkaar op (10 wordt $1 + 0 = 1$, 16 wordt $1 + 6 = 7$).
 - 3) Tel al deze cijfers samen.
In het voorbeeld dus: $2 + 6 + (1+0) + (1+4) + 2 + 6 + (1+0) + (1+4) = 28$
 - 4) Tel daarbij alle cijfers op met een oneven label.
Dus: $28 + 2 + 4 + 6 + 8 + 2 + 4 + 6 + 0 = 60$
 - 5) Indien de totale som deelbaar is door 10, dan is het gegeven nummer een geldig kredietkaartnummer.

Uitbreiding: Het laatste cijfer van het nummer wordt het controlecijfer genoemd. Breid je programma uit zodat bij een fout kredietkaartnummer, ook de juiste waarde van het controlecijfer wordt gegeven, opdat het nummer toch geldig zou zijn.

Denkvraag: Welke vaak voorkomende typfouten kunnen gedetecteerd worden?