



Programmeermethoden

Werkcollege 5: Op papier I

Het vijfde werkcollege van het vak **Programmeermethoden** vindt plaats in **zaal 302/304, zaal 303, zaal 306/308 en zalen 307 en 309** (de PC-zalen). Voor studenten Wiskunde of Informatica is dit werkcollege op **woensdag 30 september 2015, van 13:45 tot 15:30 uur**. Voor studenten Natuurkunde of Sterrenkunde, is dit werkcollege op **donderdag 1 oktober 2015, van 11:15 tot 13:00 uur**. Het wordt verzorgd door de vaste **staf**.

Voorsprek/Vragenuur in zalen 302 ... 309: dinsdag 29, woensdag 30 september, en donderdag 1 oktober 2015, van circa 15:30 tot 17:00 uur.



Voor Haagse studenten is dit werkcollege op **dinsdag 29 september 2015, 13:45-15:30 uur**.

Voorbereiding

Als voorkennis wordt bekend verondersteld:

- Inhoud colleges tot en met deze week.
- Kennis betreffende het omgaan met files. Zie onder meer dictaat, **gedeelte "aantekeningen bij de hoorcolleges"**, Hoofdstuk 3.7.

Het eigenlijke werkcollege

Maak **hiervan**:

- Maak Opgave 15 uit de **opgavenbundel**.
- Maak Opgave 17 uit de **opgavenbundel**.
- Maak Opgave 18 uit de **opgavenbundel**.
- Maak **thuis** Opgave 19 uit de **opgavenbundel**.

Doe nu achtereenvolgens deze opdrachten, die van alles met de **tweede programmeeropgave** te maken hebben:

1. Schrijf een programma dat een gegeven (invoer)file, zeg `invoer.txt`, onveranderd kopieert naar een uitvoerfile, zeg `uitvoer.txt`. Maak zelf een geschikte invoerfile als **testfile**.
Hint: dictaat, **gedeelte "aantekeningen bij de hoorcolleges"**, Hoofdstuk 3.7.
2. Laat de gebruiker voorlopig nog geen namen van in- en uitvoerfile kiezen. Bij het testen is het namelijk handig niet steeds die namen te hoeven invoeren. En noem je testfiles bijvoorbeeld `x`, dat scheelt typewerk.
3. Tel het aantal karakters in de file. Tel hierbij de regelovergangen (LineFeed: `'\n'`) ook apart.
4. Laat alle cijfers uit de file op het beeldscherm afdrukken. (Iets dergelijks is bij de programmeeropgave ook nodig.)
5. Print op het beeldscherm de kwadraten van de getallen die in de file voorkomen, één per regel. Pas daartoe het programma subtiel aan!
Bouw cijfers karakter voor karakter op tot getallen. Tip: als je getal 123 is, en je leest nu een 4 (als karakter!), hoe maak je dan het getal 1234? Gebruik dus een integer die het "huidige" getal bevat, en die meestal gewoon 0 is.
De ruwe programmastructuur is (het kan ook anders ...):

```

lees karakter in
while not einde file
    if karakter is een cijfer
        update een en ander

```

```

else
    print (kwadraat van) getal
    update van alles
    print huidig karakter
    onhoud vorig karakter (altijd handig)
    lees karakter in

```

6. Dat afdrukken gebeurt toch wel met een functie `verwerkgetal (...)`?

7. Maak geregeld (elk kwartier) een backup van je C++-file, zo nu en dan onder een andere naam!

8. Zorg ervoor dat je het bovenstaande 25 september af hebt!

=====

9. Tot nu toe was dit om er een beetje in te komen. Nu de opgave, waarvoor je het bovenstaande grotendeels kunt gebruiken. Verander steeds weinig aan je programma. In het bijzonder, voeg niet te veel get's (en put's) meer toe! Voorlopig nog niet doen: filenames kiezen.

Hint: alleen een paar eenvoudige "eigenschappen" (zoals "wat is de getalswaarde van het getal tot nu toe (vaak 0)?", en "hoeveel (mede)klinkers hebben we gehad?") zijn nodig.

10. Eerst maar het vinden van de boodschap. Let eerst nog niet op %, regelovergangen. Zorg dat het voor één regel werkt, en tel de voorbije klinkers.

11. Maak het "versleutelen". Dat lijkt op het ontcijferen. Trek random letters, net zolang je k (mede)klinkers gehad hebt, en stuur dan een letter uit de boodschap weg.

12. Doe het Lychrel-gebeuren.

Tip: schrijf een functie die het omgekeerde van het getal x teruggeeft, dus 196 wordt 691. Let op "overflow".

13. Worden regelovergangen goed afgehandeld? En het eerste karakter van de file?

14. Zorg er voor dat namen goed werken. Binnen de grote while-loop van het ontcijferen staan nu zo'n 20 tot 30 regels — of zelfs minder.

15. Maak de statistiekjes ovder de getallen in orde.

16. Zorg ervoor dat je het bovenstaande een week(je) voor de deadline af hebt!

=====

17. Gebruik alle **testfiles** om te controleren of het programma correct werkt.

18. Laat de gebruiker de namen van in- en uitvoerfile kiezen.

19. Maak nu de **tweede programmeeropgave** af. Zijn er duidelijke functies gemaakt — daar had je eerder aan moeten denken!? Heeft `main` hooguit een regel of tien?

20. Zie bij het **zesde werkcollege** hoe het verslag eruit moet zien.

Testfiles

Tip: gebruik eventueel het UNIX-commando `diff` om je eigen uitvoerfiles te vergelijken met de correcte: `diff ouwfile goedefile` moet "niks" opleveren (mits de onze goed is :-).

zie **hier**.

- Welke boodschap zit **hier** in? Tip: k is een 2-cijferig palindroom.
- En **hier** in? Let ook op de Lychrel-getallen. Tip: k is een 2-macht. Kleinste: 1; grootste: 1997; gemiddelde: 717. (Geheim: de **boodschap**.)

Huiswerk

- Maak de opgaven 18 tot en met 25 uit de **opgavenbundel**.

De antwoorden op de opgaven worden tevens via WWW verspreid, evenals de uitwerkingen van oude tentamens. Als er vragen over de opgaven zijn, stel deze dan (ook) op het college.

