

2 Invoer en Arrays

Gebruik van de commandolijn om een programma op te roepen

Eerst even wat uitleg over het runnen van je programma. Zolang je in een IDE (Integrated Development Environment) werkt, kan je een sneltoets gebruiken om je programma te laten lopen. Van zodra het programma getest en goedgekeurd is, heb je de broncode echter niet meer nodig. Enkel het programma zelf (`.exe`-extensie voor windows) en een venster met opdrachtprompt (command prompt window), volstaan. Probeer dit uit.

- Open een venster met command prompt (bvb. via de Windows startknop, tik de letters `cmd`).
- In dat venster navigeer je naar de schijf waarop je werk staat, bvb de H:-drive: tik in H: `.`
- Dan vraag je de inhoud van deze map op met het commando `dir`.
- Het commando `cd mapnaam` navigeert naar de map met de naam *mapnaam*.
- Het commando `cd ..` navigeert naar de bovenliggende map.
- Zodra je in de map bent waar het `.exe`-bestand staat, tik je de naam van dit bestand in. Dat kan zónder extensie, of met extensie `.exe`.
- Het programma loopt nu - voortijdig afsluiten kan met `Ctrl-C`.
- Dit opdrachtpromptvenster laat je open staan. Met de pijltjestoets omhoog/omlaag doorloop je alle commando's. De tab-toets vervolledigt een bestands- of mapnaam, voor zover éénduidig.

Gebruik van input/output redirection

Output redirection: je kan op de commandolijn aangeven dat de uitvoer van het programma naar een bestand moet worden weggeschreven: elk `printf`-commando stuurt de informatie niet langer naar het scherm, maar onmiddellijk naar het opgegeven bestand. Je doet dit door op de commandolijn de aanroep van het programma aan te vullen met een `>`-teken, gevolgd door (spatie en) bestandsnaam.

Een voorbeeld: de uitvoer van oef 10 wegschrijven naar het bestand `uitvoer.txt`

Typ op de commandolijn: `oef10 > uitvoer.txt`

Met `type uitvoer.txt` kan je de inhoud van het aangemaakte bestand onmiddellijk bekijken.

Input redirection: je kan aangeven dat input niet van het toetsenbord (zichtbaar op het scherm) komt, maar dat je elke input uit een bepaald bestand moet nemen. Je doet dit door op de commandolijn de aanroep van het programma aan te vullen met een `<`-teken, gevolgd door (spatie en) bestandsnaam.

Een voorbeeld:

Typ op de commandolijn: `keerom < verhaaltje.txt > omgekeerd.txt`

De programmacode in het bestand `keerom.c` zal de inlees- en uitschrijfoopdrachten `scanf` en `printf` gebruiken, maar elke `scanf`-opdracht haalt input uit het bestand `verhaaltje.txt` en elke `printf`-opdracht schrijft output weg naar het bestand `omgekeerd.txt`.

Oefening 12

Schrijf een programma dat aan de gebruiker vijf positieve gehele getallen vraagt. Geeft de gebruiker echter een negatief getal in, dan stopt het programma met getallen opvragen. Nadien schrijft het programma uit of de gebruiker inderdaad vijf positieve gehele getallen opgaf. Tot slot wordt de som van de ingegeven positieve gehele getallen uitgeschreven (ook als er niet genoeg ingegeven werden). Test uit, onder andere met de getallenreeks 1 2 3 4 -5.

Oefening 13

Schrijf voor het eerste deel van deze oefening ENKEL een hoofdprogramma, geen functies of procedures. (Houd je hier aan, of je mist de clou van de oefening.)

1. Maak in het hoofdprogramma een array aan waarin elk element een karakter is. Vul deze array bij declaratie op als volgt:

```
char letters [] = {'p','o','r','e','o','i','f','o','i','e','c','i',  
                  'i',':','a','-','t','('};
```

Bepaal de lengte van de array zonder te tellen (zie theorie)! Schrijf in het hoofdprogramma alle karakters uit die op een even positie in de array staan.

2. Schrijf nu een procedure `schrijf_even_posities(...)` die alle karakters uitschrijft die op een even positie in de gegeven array staan (Bepaal zelf aantal en aard van de parameters). Roep deze procedure op in het hoofdprogramma en test grondig uit.

Oefening 14

1. Schrijf een functie `index_van(...)`, die de (kleinste) index teruggeeft van de plaats waarop een gegeven reëel getal in een gegeven array van reële getallen gevonden wordt. Indien het getal niet aanwezig is, wordt er -1 teruggegeven. Bepaal zelf aantal en aard van de parameters.

Test uit! Declareer een array (hardgecodeerd). Lees een getal in en zoek dit getal op in de array.

2. Doe de nodige aanpassingen zodat je programma controleert of de gebruiker een reëel getal ingeeft (Indien iets wordt ingegeven dat niet aan de voorwaarden voldoet, dan blijft het programma vragen naar een reëel getal tot het ingegeven wordt).
3. Wat verandert er als de gegeven array met zekerheid geordend is? Werk deze versie uit in een tweede functie. (Deze functie test niet of dit ook effectief het geval is; het is de verantwoordelijkheid van de gebruiker om deze functie enkel voor niet-dalend gesorteerde arrays op te roepen!) Uiteraard zorg je voor een efficiënte oplossing!

Oefening 15

Schrijf een procedure die alle elementen van een gegeven array met lettertekens, één plaats naar links schuift. Het eerste karakter komt achteraan. Roep deze procedure drie keer aan op de array

```
char rij[] = {'s','a','p','a','p','p','e','l'};
```

Aangezien je de array verschillende keren moet uitschrijven, voorzie je hiervoor uiteraard een procedure.

Oefening 16

Schrijf een programma dat 20 gehele getallen genereert tussen 100 en 120 (grenzen inbegrepen) en deze ook op het scherm toont. Daarna worden alle getallen die niet gekozen werden, in stijgende volgorde uitgeschreven. Kijk kritisch na!

Heb je de grenzen en het aantal te genereren getallen in constanten bewaard?

Oefening 17

Schrijf een programma dat tekstuele input van de gebruiker, afgesloten met een \$-teken, verwerkt (zie voorwaarden na het voorbeeld!). Van elke letter wordt bijgehouden hoe dikwijls hij voorkomt. Hierbij worden hoofdletters meegerekend bij de overeenkomstige kleine letters.

Daarna worden twee histogrammen getekend (zie voorbeeld).

Voorbeeld:

Invoer van de gebruiker: Als je tevreden bent met weinig, bezit je veel. \$

- Eerste histogram:

```
a:  *
b:  **
c:
d:  *
e:  *****
...
z:  *
```

- Tweede histogram:

```

e
e
e
e
e
e
e
e
e
t
e  i  n  t
b  e  ij l n  t v
ab de g ij lmn  rst vw z
```

Voorwaarden waaraan je oplossing moet voldoen:

- Je schrijft en gebruikt drie procedures: één die de tekst inleest/verwerkt, één die een horizontaal staafdiagram tekent en één die een vertikaal staafdiagram tekent.
- Je gebruikt, buiten `scanf` en `printf`, geen bibliotheekfuncties. Zelf extra hulpfuncties schrijven kan wel zinvol zijn.

Laat nu je programma vanaf de command line draaien, waarbij je input redirection gebruikt om de input te halen uit het bestand `gandhi.txt`.