

# Inhoudsopgave

1 Inleiding.....	1
2 Toekenning- en herhalingsopdrachten (for loop) .....	2
2.1 De wet van Ohm.....	3
2.2 De spaarrekening .....	3
2.3 De transformator .....	3
3 Keuze- en herhalingsopdrachten (if, switch, while loop, do-while loop) .....	4
3.1 Kwadraat, derde macht, vierde macht en vierkantswortel .....	6
3.2 Berekeningen aan een transformator .....	7
3.3 Weerstand berekeningen van een leiding .....	8
4 Arrays en functies .....	9
4.1 Het product gedeeld door het aantal .....	10
4.2 Het verschil tussen de grootste en de kleinste .....	11
4.3 Het product gedeeld door de som .....	12
4.4 Het product van het kleinste en grootste getal. ....	13
4.5 De helft van de som van de kleinste en grootste.....	14

## 1 Inleiding

Dit hoofdstuk wordt tijdens het practicum uitgereikt!

## 2 Toekenning- en herhalingsopdrachten (for loop)

### Inleiding

In de opdrachten van dit hoofdstuk wordt gevraagd een programma te schrijven, waarin gebruik wordt gemaakt van inlees- en schrijfinstructies, toekenningsopdrachten en een herhalingsopdracht m.b.v. een for loop.

Als voorbeeld is onderstaand programma gegeven. Hierin worden o.a. wiskundige functies gebruikt. Deze zijn opgenomen in een zogenaamde mathematische bibliotheek. Enkele voorbeelden van functies die in deze bibliotheek zijn gedefinieerd zijn: sinus, cosinus, de vierkantswortel en machtsverheffen. Om deze standaard functies in jouw programma te kunnen gebruiken, zal je hierin bovenin de volgende header moeten opnemen:

```
#include <math.h>
```

De wiskundige functies die in de mathematische bibliotheek voorkomen staan achterin het boek in een bijlage aangegeven.

In onderstaand voorbeeldprogramma wordt aan de gebruiker gevraagd een positief getal  $x$  en een getal  $y$  in te voeren. Het programma berekent dan de vierkantswortel uit  $x$  en  $x$  tot de macht  $y$ . Hiervoor wordt gebruik gemaakt van de functies:  $\text{sqrt}(x)$  en  $\text{pow}(x,y)$ . De for loop zorgt ervoor dat vijf keer achter elkaar de vierkantswortel uit  $x$  en  $x$  tot de macht  $y$  wordt berekend. De eerste keer heeft  $x$  de waarde die de gebruiker heeft ingevoerd en de volgende vier keer wordt de waarde van  $x$  met 1 opgehoogd.

```
#include <stdio.h>
#include <math.h>

int main()
{
    float x, y, wortel, macht;
    int i;
    printf("Voer een positief getal x in: ");
    scanf("%f", &x);
    printf("Voer een getal y in: ");
    scanf("%f", &y);
    for (i=0 ; i<5 ; i=i+1)
    {
        wortel = sqrt(x);
        macht = pow(x, y);
        printf("De vierkantswortel uit %f is: %f\n", x, wortel);
        printf("%f tot de macht %f is: %f\n", x, y, macht);
        x=x+1;
    }
    getchar();
    getchar();
    return 0;
}
```

## 2.1 De wet van Ohm

De elektrische stroom ( $I$ ), spanning ( $U$ ), weerstand ( $R$ ) en het vermogen ( $P$ ) hebben een bepaald verband. Dit verband wordt weergegeven in de volgende twee formules:

$$U = I * R \quad \text{en} \quad P = U * I$$

Ontwerp een programma dat de gebruiker vraagt de waarden van de elektrische spanning en weerstand in te voeren. In het programma moet de weerstandswaarde vanaf de ingelezen waarde, 10 keer achter elkaar met 25 worden opgehoogd. Het programma moet voor elke afzonderlijke weerstandswaarde, beginnend vanaf de ingelezen waarde, de bijbehorende stroom en het vermogen berekenen en deze waarden telkens op het scherm afdrukken.

Maak gebruik van een for loop.

## 2.2 De spaarrekening

Wanneer je een geldbedrag op een spaarrekening hebt staan en je neemt de daarover uitgekeerde rente niet op, dan groeit het geldbedrag elk jaar met die rente.

Ontwerp een programma dat de gebruiker vraagt het beginbedrag dat op de spaarrekening staat en de rente in procenten die per jaar rente wordt uitgekeerd, in te voeren. Het bedrag staat 10 jaar op de spaarrekening. Het programma moet telkens het bedrag op de spaarrekening aan het eind van het lopende jaar berekenen, en dit bedrag samen met de rente van het betreffende jaar op het scherm afdrukken.

Maak gebruik van een for loop.

## 2.3 De transformator

De spanningen en wikkelingen van de primaire en de secundaire zijde van een transformator verhouden zich als:  $U_p / U_s = N_p / N_s$

Hierin is:

- $U_p$  = de spanning aan de primaire zijde
- $U_s$  = de spanning aan de secundaire zijde
- $N_p$  = het aantal wikkelingen aan de primaire zijde
- $N_s$  = het aantal wikkelingen aan de secundaire zijde

Ontwerp een programma dat de gebruiker vraagt, de primaire spanning ( $U_p$ ) en het aantal wikkelingen aan de primaire zijde ( $N_p$ ) en de gewenste spanning aan de secundaire zijde ( $U_s$ ) in te voeren. In het programma moet het aantal wikkelingen aan de primaire zijde ( $N_p$ ), vanaf de ingelezen waarde, 10 keer achter elkaar met 1000 ophogen. Het programma moet telkens voor elke afzonderlijke  $N_p$  - waarde, beginnend vanaf de ingelezen  $N_p$  - waarde, het aantal benodigde wikkelingen aan de secundaire zijde ( $N_s$ ) berekenen en deze berekende waarde op het scherm afdrukken.

Maak gebruik van een for loop.

### 3 Keuze- en herhalingsopdrachten (if, switch, while loop, do-while loop)

#### Inleiding

In de onderstaande voorbeeldprogramma's wordt de toegang tot het programma m.b.v. een toegangscode (een getal) beveiligd. In voorbeeld 1 wordt gebruik gemaakt van een if keuzeopdracht en in voorbeeld 2 van een for herhalingsopdracht met een geneste if keuzeopdracht. De toegangscode voor alle twee de programma's is: 9327. Voorbeeld 3 doet exact hetzelfde als voorbeeld 2 maar is iets slimmer geprogrammeerd.

#### Voorbeeld 1

De gebruiker wordt gevraagd een code in te voeren om toegang te krijgen tot het programma. Wanneer de gebruiker de toegangscode niet correct intikt wordt het programma direct beëindigd. Als de gebruiker de code wel correct intikt verschijnt er op het scherm een melding: "De code is correct!".

```
/* Bewaking met een if opdracht */
#include <stdio.h>

int main()
{
    int code;
    printf("Geef uw toegangscode: ");
    scanf("%d", &code);
    if (code==9327)
    {
        printf("De code is correct!\n");
        /* hier komt de rest van het programma */
    }
    printf("Sluit dit venster door op een toets te drukken\n");
    getchar(); getchar();
    return 0;
}
```

### Voorbeeld 2

In voorbeeld 1 wordt, als de gebruiker een verkeerde code intikt, het programma meteen beëindigd. In voorbeeld 2 wordt de gebruiker, nadat hij/zij de eerste keer een verkeerde code ingetikt, nog drie maal in de gelegenheid gesteld een nieuwe code in te tikken. Is de ingetikte code vier maal verkeerd, dan wordt het programma beëindigd. Het programma wordt verder uitgevoerd zodra de gebruiker, bij één van de vier pogingen de juiste code intikt.

```
/* Bewaking met een for en if opdracht */
#include <stdio.h>

int main()
{
    int code; int poging;
    printf("Geef uw toegangscode: ");
    scanf("%d", &code);
    for (poging=2; poging<=4; poging=poging+1)
    {
        if (code!=9327) {
            printf("De code is niet correct, probeer nogmaals: ");
            scanf("%d", &code);
        }
    }
    if (code==9327)
    {
        printf("De code is correct!\n");
        /* hier komt de rest van het programma */
    }
    printf("Sluit dit venster door op een toets te drukken\n");
    getchar(); getchar();
    return 0;
}
```

### Voorbeeld 3

Voorbeeld 3 doet exact hetzelfde als voorbeeld 2 maar is iets slimmer geprogrammeerd.

```
/* Bewaking met een for opdracht */
#include <stdio.h>

int main()
{
    int code; int poging;
    printf("Geef uw toegangscode: ");
    scanf("%d", &code);
    for (poging=2; code!=9327&& poging<=4; poging=poging+1)
    {
        printf("De code is niet correct, probeer nogmaals: ");
        scanf("%d", &code);
    }
    if (code==9327)
    {
        printf("De code is correct!\n");
        /* hier komt de rest van het programma */
    }
    printf("Sluit dit venster door op een toets te drukken\n");
    getchar(); getchar();
    return 0;
}
```

### **3.1 Kwadraat, derde macht, vierde macht en vierkantswortel**

Ontwerp een programma waarmee de gebruiker een variabele  $a$  kan invoeren en hiervan het kwadraat, de derde macht, de vierde macht of de vierkantswortel kan berekenen. Het resultaat van de berekening moet op het scherm worden afgedrukt. Voordat de gebruiker de waarde van  $a$  kan invoeren moeten eerst de mogelijkheden op het scherm worden weergegeven in de vorm van onderstaand keuzemenu.

```
**** Keuzemenu ****
1 Bereken het kwadraat van a
2 Bereken de derde macht van a
3 Bereken de vierde macht van a
4 Bereken de vierkantswortel uit a
```

Bij het kiezen uit het keuzemenu wordt gemaakt van een switch statement.

Het programma moet worden bewaakt met een toegangscode. De gebruiker mag twee maal de verkeerde code intikken; na drie maal de verkeerde code intikken moet het programma worden beëindigd. Gebruik de cijfers van je eigen postcode als toegangscode.

Als een juiste toegangscode is ingetikt moet er m.b.v. een herhalingsopdracht voor worden gezorgd dat de gebruiker van het programma ook daadwerkelijk een keuze uit het keuzemenu maakt. Met andere woorden, het programma mag niet als gevolg van een verkeerde menukeuze beëindigd worden, zonder dat er één berekening is uitgevoerd. Tip bij het testen op logische fouten: de vierkantswortel van een negatief getal bestaat niet maar de macht van een negatief getal bestaat wel.

### 3.2 Berekeningen aan een transformator

Ontwerp een programma dat de gebruiker de keuze geeft van een transformator de primaire spanning of het aantal primaire windingen of de secundaire spanning of het aantal secundaire windingen te bepalen met behulp van een keuzemenu.

Voor een transformator geldt de volgende formule:

$$U_p / U_s = N_p / N_s$$

Hierin is:  $U_p$  = de primaire spanning in Volt  
 $U_s$  = de secundaire spanning in Volt  
 $N_p$  = aantal primaire windingen  
 $N_s$  = aantal secundaire windingen

Voor het berekenen van  $U_p$  zal de gebruiker de waarden van  $U_s$ ,  $N_p$  en  $N_s$  in moeten voeren. Voor het berekenen van  $N_p$  zal de gebruiker de waarden van  $U_p$ ,  $U_s$  en  $N_s$  in moeten voeren. Voor het berekenen van  $U_s$  zal de gebruiker de waarden van  $U_p$ ,  $N_p$  en  $N_s$  in moeten voeren. Voor het berekenen van  $N_s$  zal de gebruiker de waarden van  $U_p$ ,  $U_s$  en  $N_p$  in moeten voeren. Voordat de gebruiker de benodigde gegevens kan invoeren moeten eerst de mogelijkheden op het scherm worden weergegeven in de vorm van het volgende keuzemenu.

```
**** Keuzemenu ****
1 Bereken de primaire spanning
2 Bereken het aantal primaire windingen
3 Bereken de secundaire spanning
4 Bereken het aantal secundaire windingen
```

Bij het kiezen uit het keuzemenu wordt gemaakt van een switch statement.

Het programma moet worden bewaakt met een toegangscode. De gebruiker mag twee maal de verkeerde code intikken; na drie maal de verkeerde code intikken moet het programma worden beëindigd. Gebruik de cijfers van je eigen postcode als toegangscode.

Als een juiste toegangscode is ingetikt moet er m.b.v. een herhalingsopdracht voor worden gezorgd dat de gebruiker van het programma ook daadwerkelijk een keuze uit het keuzemenu maakt. Met andere woorden, het programma mag niet als gevolg van een verkeerde menukeuze beëindigd worden, zonder dat er één berekening is uitgevoerd. Tip bij het testen op logische fouten: Variabelen die in de noemer van de formule worden gebruikt mogen niet 0 zijn maar variabelen die in de teller worden gebruikt mogen wel nul zijn.

### 3.3 Weerstand berekeningen van een leiding

De formule voor het berekenen van de elektrische weerstand van een leiding is:  $R = \frac{l r}{A}$

Hierin is:  $R$  = de weerstand van de leiding in ohm  
 $l$  = de lengte van de leiding in meters  
 $A$  = de doorsnede van de leiding in  $m^2$   
 $r$  = de soortelijke weerstand in  $\Omega m$

Ontwerp een programma dat de gebruiker de keuze geeft van een leiding de lengte ( $l$ ), de doorsnede ( $A$ ) of de weerstand ( $R$ ) te berekenen met behulp van een keuzemenu. Voor het berekenen van  $l$  zal de gebruiker de waarden van  $A$ ,  $R$  en  $r$  in moeten voeren. Voor het berekenen van  $A$  zal de gebruiker de waarden van  $l$ ,  $R$  en  $r$  in moeten voeren. Voor het berekenen van  $R$  zal de gebruiker de waarden van  $A$ ,  $l$  en  $r$  in moeten voeren. Voordat de gebruiker de benodigde gegevens in kan voeren moeten eerst de mogelijkheden op het scherm worden weergegeven in de vorm van onderstaand keuzemenu:

```
**** Keuzemenu ****  
1 Bereken de lengte  
2 Bereken de doorsnede  
3 Bereken de weerstand
```

Bij het kiezen uit het keuzemenu wordt gemaakt van een switch statement.

Het programma moet worden bewaakt met een toegangscode. De gebruiker mag twee maal de verkeerde code intikken; na drie maal de verkeerde code intikken moet het programma worden beëindigd. Gebruik de cijfers van je eigen postcode als toegangscode.

Als een juiste toegangscode is ingetikt moet er m.b.v. een herhalingsopdracht voor worden gezorgd dat de gebruiker van het programma ook daadwerkelijk een keuze uit het keuzemenu maakt. Met andere woorden, het programma mag niet als gevolg van een verkeerde menukeuze beëindigd worden, zonder dat er één berekening is uitgevoerd. Tip bij het testen op logische fouten: Variabelen die in de noemer van de formule worden gebruikt mogen niet 0 zijn maar variabelen die in de teller worden gebruikt mogen wel nul zijn.



## 4 Arrays en functies

### Inleiding

De opgaven in deze serie gaan over het gebruik van ééndimensionale array's en het gebruik van functies. Elke opdracht uit dit hoofdstuk bestaat uit een deel A en een deel B.

#### Deel A

In deel A zal een programma moeten worden geschreven waarin géén functies mogen voorkomen. Het programma moet voldoen aan eisen die in deze inleiding en in de opdracht worden gesteld.

Bij elke opdracht geldt dat er alléén positieve gehele getallen in de array mogen worden ingevoerd. Het aantal getallen dat in de array ingelezen wordt, ligt niet van te voren vast, maar wordt door de gebruiker bepaald. De array heeft echter wel een vaste grootte, die tijdens declaratie wordt vastgelegd. Het invoeren van getallen in de array wordt door de gebruiker gestopt wanneer deze de waarde 0 invoert. De waarde 0 mag niet in de array worden opgeslagen.

Let op! Er moet ook getest worden of het aantal van de ingelezen getallen de lengte van het array niet overschrijdt.

#### Deel B

In het deel B moet het programma van deel A worden opgesplitst in een hoofdprogramma en een functie. De header van de functie staat al in de opdracht aangegeven. Het programma zal zodanig moeten worden aangepast dat er voldaan wordt aan de in deel B gestelde opdracht. Bij de functie moet een afzonderlijke PSD met bijbehorende variabelen-declaratietabel worden opgesteld.

Voordat programma van deel B ter goedkeuring worden aangeboden, zal eerst deel A voldoende moeten zijn.

## 4.1 Het product gedeeld door het aantal

Lees eerst aandachtig de inleiding van dit hoofdstuk op pagina 9!

- A. Ontwerp een programma waarmee eerst een array gevuld wordt met willekeurige positieve gehele getallen, die de gebruiker van het programma invoert. Als de gebruiker per ongeluk een negatief getal invoert dan moet een foutmelding worden gegeven en de gebruiker moet de kans krijgen om opnieuw een getal in te voeren. Nadat de getallen zijn ingevoerd, worden deze getallen met elkaar vermenigvuldigd en gedeeld door het aantal ingevoerde getallen. Druk de ingevoerde getallen en het resultaat van de berekening af op het scherm. Bijvoorbeeld:

```
De ingevoerde getallen zijn: 2, 1, 10, 7  
Het resultaat is: 35
```

- B. Splits het programma op in een hoofdprogramma en een functie. De functie moet in staat zijn, om de getallen die in de array staan met elkaar te vermenigvuldigen en het resultaat hiervan te delen door het aantal getallen die in de array staan.

De header van de functie dient er als volgt uit te zien:

```
float produktgem(int A[], int aantal)
```

Hierin zijn de formele argumenten:

- A (de array).
- aantal (het aantal elementen van de array dat gevuld is)

Druk in het hoofdprogramma de getallen van de array en het getal dat de functie teruggeeft af op het scherm.

## 4.2 Het verschil tussen de grootste en de kleinste

Lees eerst aandachtig de inleiding van dit hoofdstuk op pagina 9!

- A. Ontwerp een programma waarmee eerst een array gevuld wordt met willekeurige positieve gehele getallen, die de gebruiker van het programma invoert. Als de gebruiker per ongeluk een negatief getal invoert dan moet een foutmelding worden gegeven en de gebruiker moet de kans krijgen om opnieuw een getal in te voeren. Nadat de getallen ingevoerd zijn, moet uit de array het verschil tussen het grootste en het kleinste getal bepaald worden. Druk de ingevoerde getallen en het resultaat van de berekening af op het scherm. Bijvoorbeeld:

```
De ingevoerde getallen zijn: 9, 3, 7, 10  
Het resultaat is: 7
```

- B. Splits het programma op in een hoofdprogramma en een functie. De functie moet in staat zijn, om het verschil tussen het grootste en het kleinste getal te bepalen van alle getallen die in de array staan.

De header van de functie dient er als volgt uit te zien:

```
int verschgrootklein(int A[], int aantal)
```

Hierin zijn de formele argumenten:

- `A` (de array).
- `aantal` (het aantal elementen van de array dat gevuld is)

Druk in het hoofdprogramma de getallen van de array en het getal dat de functie teruggeeft af op het scherm.

### 4.3 Het product gedeeld door de som

Lees eerst aandachtig de inleiding van dit hoofdstuk op pagina 9!

- A. Ontwerp een programma waarmee eerst een array gevuld wordt met willekeurige positieve gehele getallen, die de gebruiker van het programma invoert. Als de gebruiker per ongeluk een negatief getal invoert dan moet een foutmelding worden gegeven en de gebruiker moet de kans krijgen om opnieuw een getal in te voeren. Nadat de getallen zijn ingevoerd, worden deze getallen met elkaar vermenigvuldigd en gedeeld door de optelling van de getallen in de array. Druk de ingevoerde getallen en het resultaat van de berekening af op het scherm. Bijvoorbeeld:

```
De ingevoerde getallen zijn: 3, 9, 7, 10  
Het resultaat is: 65.17
```

- B. Splits het programma op in een hoofdprogramma en een functie. De functie moet in staat zijn, om de getallen die in de array staan met elkaar te vermenigvuldigen en te delen door de som van de getallen die in de array staan.

De header van de functie dient er als volgt uit te zien:

```
float produktdeel(int A[], int aantal)
```

Hierin zijn de formele argumenten:

- `A` (de array).
- `aantal` (het aantal elementen van de array dat gevuld is)

Druk in het hoofdprogramma de getallen van de array en het getal dat de functie teruggeeft af op het scherm.

#### **4.4 Het product van het kleinste en grootste getal.**

Lees eerst aandachtig de inleiding van dit hoofdstuk op pagina 9!

- A. Ontwerp een programma waarmee eerst een array gevuld wordt met willekeurige positieve gehele getallen, die de gebruiker van het programma invoert. Als de gebruiker per ongeluk een negatief getal invoert dan moet een foutmelding worden gegeven en de gebruiker moet de kans krijgen om opnieuw een getal in te voeren. Nadat de getallen zijn ingevoerd, wordt het grootste en kleinste getal uit de array met elkaar vermenigvuldigd. Druk de ingevoerde getallen en het resultaat van de berekening af op het scherm. Bijvoorbeeld:

```
De ingevoerde getallen zijn: 5, 2, 11, 13  
Het resultaat is: 26
```

- B. Splits het programma op in een hoofdprogramma en een functie. De functie moet in staat zijn, om het kleinste en grootste getal uit de array met elkaar te vermenigvuldigen.

De header van de functie dient er als volgt uit te zien:

```
int prodgrootklein(int A[], int aantal)
```

Hierin zijn de formele argumenten:

- A (de array).
- aantal (het aantal elementen van de array dat gevuld is)

Druk in het hoofdprogramma de getallen van de array en het getal dat de functie teruggeeft af op het scherm.

## 4.5 De helft van de som van de kleinste en grootste

Lees eerst aandachtig de inleiding van dit hoofdstuk op pagina 9!

- A. Ontwerp een programma waarmee eerst een array gevuld wordt met willekeurige positieve gehele getallen, die de gebruiker van het programma invoert. Als de gebruiker per ongeluk een negatief getal invoert dan moet een foutmelding worden gegeven en de gebruiker moet de kans krijgen om opnieuw een getal in te voeren. Nadat de getallen zijn ingevoerd, wordt het grootste en kleinste getal van alle getallen uit de array bij elkaar opgeteld en gedeeld door 2. Druk de ingevoerde getallen en het resultaat van de berekening af op het scherm. Bijvoorbeeld:  
De ingevoerde getallen zijn: 5, 2, 11, 13  
Het resultaat is: 7.5
- B. Splits het programma op in een hoofdprogramma en een functie. De functie moet in staat zijn, om het kleinste en grootste getal van alle getallen uit de array bij elkaar op te tellen en dit te delen door 2.

De header van de functie dient er als volgt uit te zien:

```
float halfkleingroot(int A[], int aantal)
```

Hierin zijn de formele argumenten:

- A (de array).
- aantal (het aantal elementen van de array dat gevuld is)

Druk in het hoofdprogramma de getallen van de array en het getal dat de functie teruggeeft af op het scherm.