



*CPROUC/2021-2022*

Jesse op den Brouw

**CPROUC**

Beslissen en herhalen

**DE HAAGSE**  
HOGESCHOOL

# Beslissen

- Er zijn drie soorten beslissingen
- if ...
- if ... else ...
- switch

# If

- Met if kunnen we statements uitvoeren als de voorwaarde waar is.

```
if (a > b)
{
    /* Statements worden uitgevoerd als voorwaarde waar is */
}
```

```
if (a > 0 && b > 0)
{
    /* Statements worden uitgevoerd als voorwaarde waar is */
}
```

## If else

- Met if else kunnen we statements uitvoeren als de voorwaarde waar of niet waar is.

```
if (a > b)    /* bepaal het maximum van a en b */
{
    max = a;
}
else
{
    max = b;
}
```

# Switch

- Met switch kunnen we een variabele testen op meerdere gehele constanten. Statements worden uitgevoerd bij de keuze die waar is. Default als geen van de keuzes waar zijn. Break is nodig om de switch te verlaten.

```
switch (a)
{
    case 1: printf("een"); break;
    case 2: printf("twee"); break;
    case 3: printf("drie"); break;
    default: printf("iets anders"); break;
}
```

# Switch – fall through

- Switch met fall through. Statements bij 3 ook uitgevoerd bij 1 en 2

```
switch (a)
{
    case 1:
    case 2:
    case 3: printf("een, twee of drie"); break;
    default: printf("iets anders"); break;
}
```

# Herhalen

- Er zijn drie soorten herhalingen
- for
- while
- do while

# For

- For wordt gebruikt als het aantal herhalingen bekend zijn:

```
for (initialisatie; voorwaarde; aanpassen)
```

- Voorbeeld: 10x herhalen

```
int i;  
for (i = 1; i < 11; i = i + 1)  
{  
    printf("%d ", i);  
}
```



# For

- For mag ook “omlaag”
- Voorbeeld: 5x herhalen

```
int i;  
for (i = 10; i > 0; i = i - 2)  
{  
    printf("%d ", i);  
}
```

# While

- While wordt gebruikt als het aantal herhalingen *0* of meer zijn
- Voorbeeld:

```
while (a > b) /* herhaal zolang a > b */  
{  
    printf("a is groter dan b\n");  
    b = b + 1;  
}
```

## Do while

- Do while wordt gebruikt als het aantal herhalingen **1** of meer zijn
- Voorbeeld inlezen positief getal:

```
int getal;
do {
    printf("Geef getal groter dan 0: ");
    scanf("%d", &getal);
} while (getal < 1); /* herhaal zolang a < 1 */
```

# For – while equivalentie

Het for statement met

```
for (initialisatie; voorwaarde; aanpassen) { ... }
```

is equivalent met (doet hetzelfde als):

```
initialisatie;  
while (voorwaarde) {  
    ...  
    aanpassen;  
}
```

# Accolades gebruiken!

- Gebruik altijd accolades ook al is er maar één statement.

```
if (a > b)    /* bepaal het maximum van a en b */  
    max = a;  
else  
    max = b;
```

- Dit gaat verkeerd als er meer statements worden toegevoegd.

## Voorbeeld – tafel van ...

```
#include <stdio.h>

#pragma warning(disable: 4996) /* voor Visual Studio */

int main(void) {

    int i, tafel;

    do {
        printf("Geef getal tussen 1 en 10: ");
        scanf("%d", &tafel);
    } while (tafel < 1 || tafel > 10);

    for (i = 1; i < 11; i = i + 1) {
        printf("%3d x %3d = %3d\n", i, tafel, i * tafel);
    }
}
```

# Eeuwig durende herhaling

- Met for

```
for (;;)
{
}
```

- Met while

```
while (1)
{
}
```

# Flowcharts

- Een flowchart is een grafische afbeelding van een (deel van een) programma. Hiervoor worden symbolen gebruikt. Pijlen verbinden de symbolen.



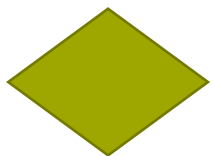
startpunt, eindpunt



statement



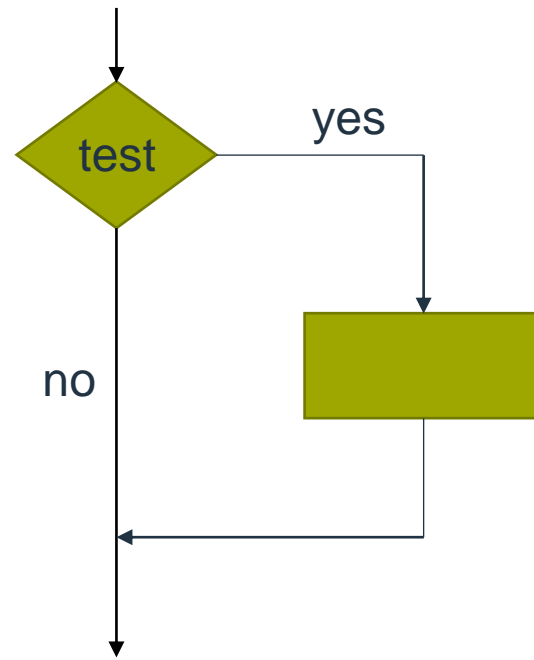
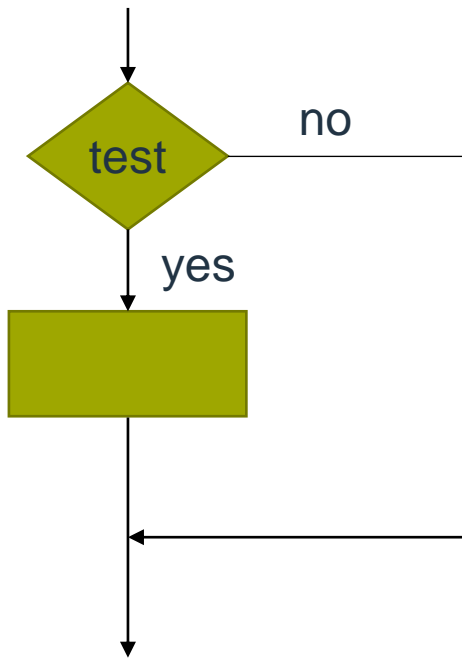
inlezen, afdrukken



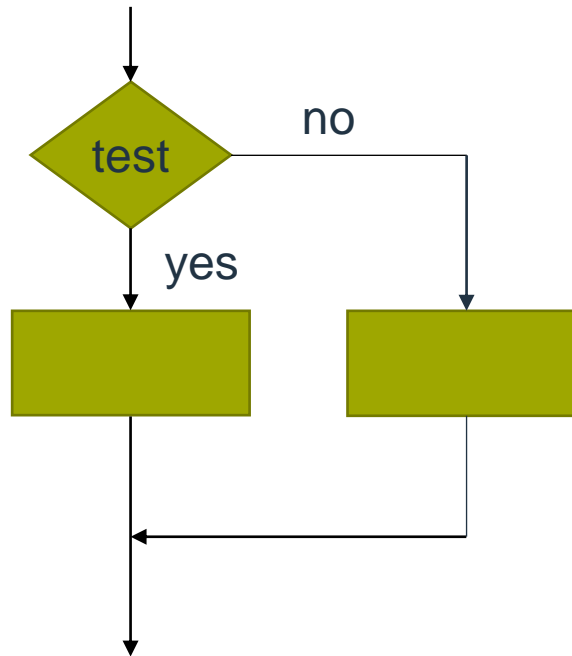
beslissen (alleen waar, niet waar)



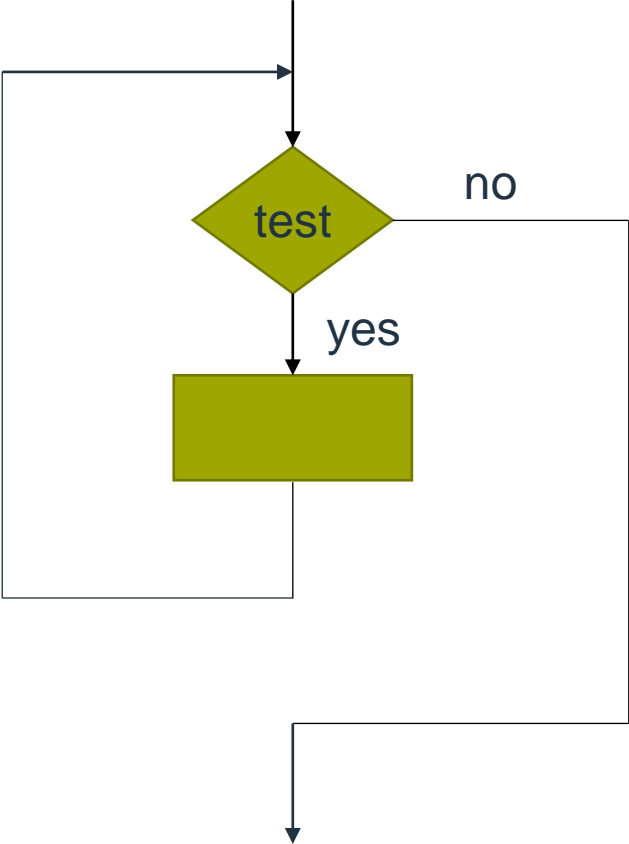
# If



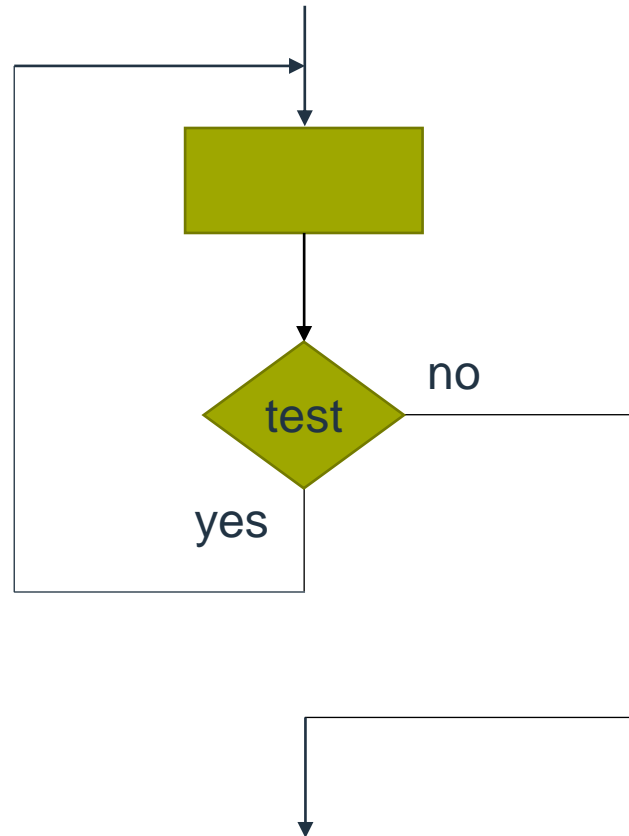
# If else



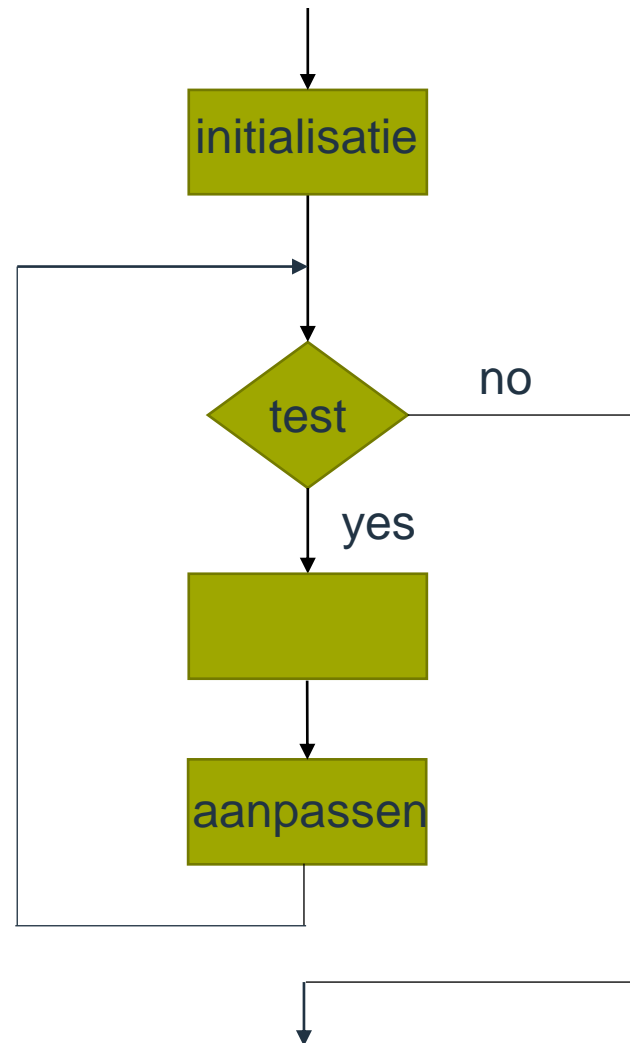
# While



# Do while

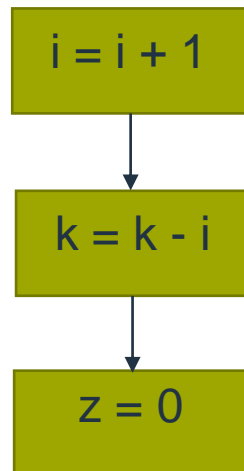


# For (via while-equivalentie)



## Schrijf compact

- Gebruik niet meer dan 15 symbolen op één A4-tje.



$i = i + 1$   
 $k = k - i$   
 $z = 0$

**let's change**