

Løkker

Løkker brukes til å gjenta et sett med instruksjoner. Antall instruksjoner og antall gjentakelser kan variere. Instruksjonene som skal gjentas omslutes med et sett løkkeparenteser og kan på mange måter betraktes som egen blokk eller underprogram inne i hovedprogrammet. Vi går i tur og orden igjennom de tre løkketyperne **while**-løkke, **for**-løkke og **do-while** løkke.

WHILE - LØKKA

```
while (betingelse)
{
    instruksjoner;
}
```

Instruksjonene inne i løkkeparentesen gjentas så lenge som betingelsen er oppfylt. Dette vil i praksis si at **antall gjennomløp ikke er forhåndsbestemt**, men avhengig av for eksempel inntastede data eller beregninger inne i løkka. Samtidig med at dette gir en mer fleksibel løkke, er det også en type løkke som kan ende i katastrofe: Hvis du som programmerer ikke sørger for at betingelsen en eller annen gang ikke blir oppfylt, vil du ha laget en uendelig løkke! Da kan det tenkes at du må slå av og på maskinen igjen for å komme ut av det og derfor **må** du sørge for å lagre programmet ditt **før** du prøvekjører det.

Skriv inn og kjør dette programmet:

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

int main()
{
    int teller=0;
    while (teller<10)
    {
        teller++;
        printf("%d\n", teller );
    }
    system("pause");
    return 0;
}
```

Tenk litt over hva som kan skje hvis jeg ikke setter teller=0 før løkka starter. Du må gjerne forsøke å endre på verdier i programmet og se hva som skjer. Løkker brukes ofte til å summere tall. Prøv dette programmet.

```
#include <stdio.h>
#include <stdlib.h>
//-----

int main()
{
    int tall;
    int sum=0;
    int svar=1;
    while (svar==1)
    {
        printf("Tast inn et heltall: ");
        scanf("%d",&tall);
        sum=sum+tall;
        printf("\nSummen er nå %d ",sum);
        printf("\nVil du fortsette ? Tast 1 for ja eller 2 for nei.");
        scanf("%d",&svar);
    }
    return 0;
}
```

For-løkke

Her er et eksempel på bruk av **for**-løkke i stedet for **while**-løkke (se det første eksempelet).

```
#include <stdio.h>
#include <stdlib.h>
//-----

int main(int argc, char **argv)
{
    int teller;
    for (teller=1;teller<10;teller++)
    {
        printf("%d\n", teller );
    }
    system("pause");
    return 0;
}
```

Generelt ser **for**-løkka slik ut:

for (startverdi;betingelse;endring)

```
{
    instruksjon1;
    instruksjon2;
    instruksjon3;
    :
    :
}
```

Normalt "styres" **for**-løkka av en heltallsvariabel som gis en startverdi, som det settes en betingelse for og som gis en fast endring for hver gjentakelse av løkka. Denne

endringen utføres etter at innholdet i blokken er utført. Vil du lage ei **for**-løkke som gjentas 100 ganger, kan dette gjøres på følgende måte ved hjelp av en heltallsvariabel i:

```
for (i=1;i<=100;i++)
{
:
}
```

for-løkka brukes mest i situasjoner der du vil ha **et bestemt antall gjennomløp** av løkka.

Oppgaver. Her kan du bruke for-løkke

- Skriv et program (program9) som summerer alle heltallene fra 1 til **n**. Tallet **n** skal du spørre bruker om og testes inn fra tastaturet før summeringen begynner. Summen av tallene skrives ut til slutt.
- Skriv et program (program10) som leser inn 5 tall fra tastaturet, beregner summen av tallene og regner ut gjennomsnittet.
- Du skal skrive et program (program11) som beregner hvor mye et beløp vokser på en bankkonto når beløp, rentefot og antall år beløpet skal stå på kontoen blir oppgitt fra tastaturet. Hint: hvordan ser renteformelen ut?

Oppgave. Prøv while-løkke her

Lag et program som leser inn nettoprisen for en rekke varer. For hver innlest nettopris skal følgende beregningsoppsett skrives ut på skjermen:

```
Nettopris xxxxx.xx kr
+ 23% mva xxxxx.xx kr
-----
Bruttopris xxxxx.xx kr
```

Programmet skal avslutte når en innlest nettopris er negativ. I utskriften er det ønsket 2 desimaler og måten du skriver ut et %-tegn er å sette 2 %-tegn i teksten i printf kommandoen.

Litt vanskeligere oppgave.

Skriv et program (program12) som ber deg taste inn positive heltall fra tastaturet og som tilslutt skriver ut det minste tallet du tastet inn. Bruk et negativt heltall som signal på at du vil avslutte.

Løkker - del 2.

do -while løkken.

syntaks:

do

instruksjon; *// eller { flere instruksjoner }*

while (logisk uttrykk /betingelse);

do-while -løkken brukes når antall gjennomløp er **minst én gang**. Betingelsen skal være oppfylt for at et nytt gjennomløp av løkken blir utført. Betingelsen sjekkes altså etter at instruksjon er utført en gang.

Oppdatering av løkkebetingelsen skal være inkludert i instruksjonene.

Eksempel 1:

I 2. kolonne i tabellen nedenfor, finner du det eksemplet som du har sett på med while - løkke.

I 1.kolonne er det eksemplet omskrevet med do-while -løkke. Forskjellen mellom disse to, er at svar-variablen må ha verdi før du kommer til while - løkken. Mens do while - løkken, blir gjennomført minst én gang uavhengig hvilken verdi svar-variabelen har.

do - while	while
<pre>#define _CRT_SECURE_NO_WARNINGS #include <stdio.h> #include <stdlib.h> int main() { int tall; int sum = 0; char svar; do { printf("Tast inn et heltall: "); scanf("%d", &tall); sum = sum + tall; printf("\nSummen er nå : %d ", sum); printf("\nVil du fortsette (j/n) ?"); scanf(" %c", &svar); } while (svar == 'j'); system("pause"); return 0; }</pre>	<pre>#define _CRT_SECURE_NO_WARNINGS #include <stdio.h> #include <stdlib.h> int main() { int tall; int sum = 0; char svar='j'; while (svar=='j') { printf("Tast inn et heltall: "); scanf("%d", &tall); sum = sum + tall; printf("\nSummen er nå : %d ", sum); printf("\nVil du fortsette (j/n) ?"); scanf(" %c", &svar); } system("pause"); return 0; }</pre>

Char – Karakter variabler.

Char er en type variabel som kan inneholde tegn i form av bokstaver og tall.

NB !

Karakteren 'A' - er en karakter (char)

Karakteren '£' - er en karakter (char)

Karakteren '1' - er en karakter (char)

Tallet 1 - er en integer (int) (legg merke til at det ikke er benyttet tegnet ' (enkel apostrof). Dette skiller tallet 1 fra tegnet 1. Variabler av type char deklarerer slik:

char svar; (Se eksempel 1 over.)

Eksempel 2:

Les gjennom programmet. Prøv og finne ut følgende:

- Hva utfører dette programmet?
- Hva er betingelsen for at programmet vil bli avsluttet?
- Hvor blir oppdatering av betingelsen gjennomført?

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

//-----

int main()
{
    int i;                      /* i blir ofte brukt som tellervariabel */
    int multiplikator = 2;
    char svar;
    do
    {
        for ( i=1; i<=10; i++)
            printf("\n %d * %d = %d",i,multiplikator,i*multiplikator);
        /* De to linjene ovenfor sørger for at du får en multiplikasjonstabell */
        printf("\nSkal jeg skrive ut den neste tabellen ?");
        scanf( " %c", &svar );
        getchar( ); /*Fjerner linjeskift */
        multiplikator++;
    } while(svar == 'j' || svar == 'J');

    return 0;
}
```

Repetisjonsoppgaver:

- (program13). Benytt while -løkken istedenfor do while på eksempl 2 ovenfor.
- (program14). Omskriv eksemplet ovenfor slik at programmet leser inn hvilken multiplikasjonstabell som skal skrives ut.

Noe vanskeligere:

- (program15). Omskriv programmet slik at du leser inn en multiplikator, n, og deretter skriver programmet ut alle tabellene mellom 2 og n (fra og med 2 - til og med n).
- (program16). Omskriv dette programmet videre slik at programmet leser inn to multiplikatorer n og m og deretter skriver programmet ut alle tabellene mellom n og m (fra og med n - til og med m).

Oppgave:

Skriv et program (program17) som gir følgende meldinger på skjermen:

---- Informasjon om studieretningene ved HIVE ----

D: DA

E: EN og EA

M: MT

P: PD

Tast inn ett av tegnene i venstre kolonne for mer informasjon:

Bruker skal ved å taste inn en av de oppgitte bokstaver få en av følgende hjelpetekster:

D Studieretning datateknikk

Hoegskolelektor: Helge Herheim

E Studieretning elektronikk

Hoegskolelektor: Rune Langøy

eller

Studieretning maritime elektro-automasjon

Hoegskolelaerer: Ragnar N. Berg

M Studieretning mikrosystemteknologi

Foersteamenuensis: Kristin Imenes

P Studieretning produktdesign

Hoegskolelektor: Birgitte Prebensen

Skjermen skal blankes før hjelpe-tekst skrives ut på skjermen.

(Hint: system("cls");)