

## Kapitel 4 Loopar

## Övning 1: Loopar och For(...) -loopen.

I denna första aktivitet för kapitel 4 ska du lära dig att använda loopar och strukturen och användningen av For(...) -loopen.

**Lärarkommentar:** Det finns tre typer av loopar i TI-Basic: **For**, **While**, och **Repeat**. I en loopstruktur kan ett program processa en uppsättning av satser igen och igen, antingen genom upprepning (iteration) med en följd av värden (som i **For**-loopen) eller tills ett speciellt villkor är uppfyllt eller inte uppfyllt (som i **While** och **Repeat**). I detta kapitel så handlar varje övning bara om en av dessa strukturer.

Det är också möjligt att använda de mer ålderdomliga **Lbl** och **Goto**-satser för att göra en loop men det leder till ofta till dåliga vanor och kan resultera i programfel och om det inte görs korrekt. Vi undviker därför referenser till **Lbl** och **Goto**. Emellertid så används **Lbl**-satsen ibland och den är faktiskt nödvändig tillsammans med en **Menu**-sats när man designar anpassade menyer i ett program. För villkorssatser och loopar behövs dock inte någon **Goto**-sats.

Program kan bli komplicerade eftersom det ofta är nödvändigt att blanda in alla kontrollstrukturer (If-satser och loopar) i ett program för att kunna arbeta med mer komplexa algoritmer. Det är ju detta som gör programmering så spännande och intressant ... och roligt!

### Loopar

En loop är en metod att repetera ett antal satser. Alla programspråk har åtminstone en loopstruktur. **Loopstrukturen innebär ett sätt att gå bakåt i ett program till en tidigare plats.** TI-Basic har tre olika typer av loopar. En oändlig loop har inget slut!

För att avbryta en programkörning tryck på **[on]**. Du får då två möjligheter att gå vidare: *Avsluta* eller *Gå till*. Med *Avsluta* så återgår du till räknarens startskärm och med *Gå till* hamnar du i programeditorn till det ställe där programmet stoppades.

De tre TI-Basic-looparna är:

**For( ) ... End**

**While <villkor är sant> ... End**

**Repeat <tills villkor är sant> ... End**

I fortsättningen i denna aktivitet tar vi bara upp **For( )**-loopen.

#### The For( ... ) Loop

Struktur: **For**(variabel, startvärde, slutvärde)  
           loopkropp  
           **End**

Exempel: **For**(A,1,10)

**Disp** A           ← loopkropp  
**End**

#### Syfte:

- Förstå loopar.
- Använda **For(...)**-loopen för att skapa en lista med värden

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: OÄNDLIG
:While 1>0
:Disp "TILL OÄNDLIGHET OCH
BORTOM"
:End
:█
```

```
NORMAL FLYT AUTO REELL RAD MP
CTL I/O FÄRG EXEK HUB
1: If
2: Then
3: Else
4: For(
5: While
6: Repeat
7: End
8: Pause
9: Lbl1
```

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: TILL10
:For(A,1,10)
:Disp A
:End
:
```

**Obs:**

**For( )**-satsen kräver en variabel (loopens kontrollvariabel), ett startvärde och ett slutvärde. Man separerar med kommatecken. Start- och slutvärdena kan vara variabler. Loopkroppen kan bestå av så många satser som behövs men man får inte ändra loopens kontrollvariabel. Loopen körs från startvärdet till slutvärdet med steglängden 1.

**Lärarkommentar:** Loopen slutar när variabeln överskrider sitt slutvärde. Om du lägger till **Disp A** när loopen i programmet slutar så ser man att värdet är 11, inte 10.

```
NORMAL FLYT AUTO REELL RAD MP
3
4
5
6
7
8
9
10
..... Klar..
```

**For-loop med stegvis ökning/minskning som inte är 1**

Det finns ett valfritt fjärde argument för **For( )**-satsen: *steget*. *Steget*, det värde med vilket loopens kontrollvariabel ökar/minskar vid varje upprepning (iteration) hos loopen. Det förvalda värdet är 1.

**Lärarkommentar:** Om loop-steget är negativt så slutar loopen när loop-variabeln är *mindre* än slutvärdet.

**For(A,1,10,3)** startar med **A=1**, sedan läggs 3 till **A** varje gång som loopen upprepas. Loopen stannar när **A** är *större* 10. Steget kan vara också vara ett negativt tal.

**For(B,10,0,-1)** räknar *ner* från 10 till 0.

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM:TILL10
:For(A,1,10,3)
:Disp A
:End
:
```

```
NORMAL FLYT AUTO REELL RAD MP
Pr9mTILL10
1
4
7
10
..... Klar..
```

**Programmering med For(...)**

Låt oss skriva ett program som visar en tabell av tal och deras kvadrater. Användaren ska mata in de undre och övre gränserna. Den trixiga delen är att visa (**Disp**) de två talen (talet och dessa kvadrat) på samma rad. Man kan då använda *listor*.

**Obs:**

U och O används för att representera Undre och Övre. Å, Ä och Ö saknas ju på räknaren. For( )-satsen använder värdena hos U och O.

Klammerparenteserna, som används för listor, når du genom att trycka  $\boxed{2nd}\boxed{\{}$  och  $\boxed{2nd}\boxed{\}}$ .

**Lärarkommentar:** Att visa en lista är en behändig metod att visa mer än ett värde på en rad på startskärmen (grundfönstret).

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM:KVADRAT
:ClrHome
:Input "UNDRE ",U
:Input "ÖVRE ",O
:For(A,U,O)
:Disp {A,A^2}
:Pause
:End
:
```

Kör programmet och skriv in den undre och övre gränsen för tabellen. Om listorna dyker upp och försvinner för snabbt så kan du lägga till en **Pause**-sats efter **Disp**-satsen och innan **End**.

## 10 Minutes of Code

TI-84 Plus-familjen

## KAPITEL 4: ÖVNING 1

LÄRARKOMMENTARER

### Utmaning:

Använd en **If ... Then... End** struktur för att **Pausa** efter vart 5:e par av tal.

### Förslag på kod:

```
:If (A-U+1)/5=heltal((A-U+1)/5)
```

```
:Pause "Tryck Enter"
```