

Kapitel 3: Villkorssatser

Övning 2: If...Then...End och sammansatta villkor

I denna andra lektion för enhet 3 kommer du att lära dig om en mycket bättre villkorsstruktur och sammansatta villkor.

Syfte:

- Undersöka If...Then...End-struktur.
- Skapa sammansatta villkor med logiska operatörer.
- Skriva ett program med If...Then...End-struktur som undersöker områden i koordinatplanet.

The If...Then...End struktur

TI Basic har en unik **If...Then**-struktur som utnyttjar nyckelordet **End** för att styra satserna som formar kodblocket och som kommer att processas när villkoret är sant. Så här ser det ut:

```

If <villkor>
  Then
    <sant block: genomför dessa satser när <villkor> är sant
  End

```

Obs:

If följs av ett <villkor>.

Then finns omedelbart under **If**, på egen rad.

Det finns en eller flera satser i <sant block>.

End indikerar slutet på **Then**-blocket och satserna under **End** kommer att processas.

End är inte slutet på programmet! Det är slutet på **If...Then...End**-strukturen.

Lärarkommentar: Denna sats är att föredra framför den tidigare **If**-satsen eftersom den är lättare att läsa. Blocket kan vara en enstaka sats (eller till och med ingen sats alls!). **Then** och **End** har sina egna rader i programmet. Blocket kan också innehålla en annan **If**-sats. Varje **If Then** har sedan ett motsvarande **End**. Vi tar upp detta med nästlade strukturer senare.

Sammansatta villkor

Sammansatta villkor innehåller mer än ett relationsuttryck. De logiska operatorerna **och**, **eller**, **xeller** and **inte** (finns i testmenyn. Tryck $\boxed{2nd}\boxed{test}$) och välj LOGIK. Dessa operatörer ger dig möjlighet att bygga sammansatta villkor.

Exempel:

- $X > 0$ **och** $Y > 0$ är sant när både X och Y är positiva
- $X > 0$ **eller** $Y > 0$ är sant när antingen X eller Y är positiva (eller båda)
- **inte**($X > 0$ **och** $Y > 0$) är sant när varken X eller Y är inte positiva.
Det betyder samma sak som **X** ≤ 0 **or** **Y** ≤ 0
- $X > 0$ **xeller** $Y > 0$ är sant när X eller Y är positiva *men inte båda*.
Det betyder samma sak som... **X** > 0 **eller** **Y** > 0 **och** **inte**(**X** > 0 **och** **Y** > 0)

xeller står för "exklusivt" eller och är sant när någon del är sann men inte båda delarna. Du kan inte "knyta ihop" relationsoperatörer: $2 < A < 3$ ska tolkas som

```

NORMAL FLYT AUTO REELL RAD MP
PROGRAM: IFTHEN
:Input
:If X>0 och Y>0
:Then
:Disp "1:A KVADRANTEN"
:Disp "X POSITIVT"
:Disp "Y POSITIVT"
:End
:

```

```

NORMAL FLYT AUTO REELL RAD MP
TEST LOGIK
1:och
2:eller
3:xeller
4:inte(

```

10 Minutes of Code

TI-84 Plus-familjen

KAPITEL 3: ÖVNING 2

LÄRARKOMMENTARER

“A är mellan 2 och 3” och måste kodas som **2<A och A<3**. De logiska operatorerna har en prioritetsordning precis som de aritmetiska operatorerna +, -, *, och /.

A<0 eller A<5 och A>2 betyder att A kan vara negativt eller mellan 2 och 5 **och** processas före **eller** (precis som multiplikation går före addition).

Lärarkommentar: Lite övning på att utvärdera logiska villkor kan kanske behövas här. Fylla i sanningstabeller kan vara en bra övning.

A	B	A och B	A eller B	inte(A)
Sant	Sant	Sant	Sant	Falskt
Sant	Falskt	Falskt	Sant	Falskt
Falskt	Sant	Falskt	Sant	Sant
Falskt	Falskt	Falskt	Falskt	Sant

Programmering med If...Then...End-satser

Försök nu köra programmet IFTHEN till höger.

Obs: **Input** har ingen variabel. Detta är en speciell funktion i TI-Basic. Du kanske kommer ihåg från kapitel 2 att GRAF-fönstret visas och du kan röra markören fritt och trycka på **enter** för att lagra värden för **X** och **Y** (koordinaterna).

och finns i LOGIK- menyn. Tryck **[2nd][test]**.

Then finns på en egen rad precis efter **If**-satsen.

End är slutet på 'sant-blocket (uppsättningen av satser som körs när villkoret är sant). Det är inte slutet på själva programmet.

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: IFTHEN
:Input
:If X>0 och Y>0
:Then
:Disp "1:A KVADRANTEN"
:Disp "X POSITIVT"
:Disp "Y POSITIVT"
:End
:Disp "SLUT!"
:█
```

Skri­va färdigt programmet

I ett graffönster finns det flera områden med speciella namn. Vi har de fyra kvadranterna och positiva respektive negativa x- och y-axlar. Vi ska nu skriva ett program som låter användaren markera en punkt i ett koordinatsystem på GRAF-skärmen och sedan låta programmet bestämma var punkten ligger.

Vi har nedan börjat skriva på programmet. Försök nu att slutföra det.

Input Obs: ingen variabel här!

Disp X,Y

If X>0 och Y>0

Then

Disp "1:A KVADRANTEN"

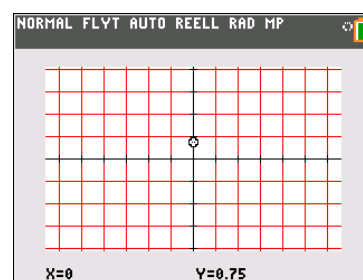
End

If X=0 och Y>0

Then

Disp "POSITIVA Y-AXELN"

End



```
NORMAL FLYT AUTO REELL RAD MP
PrgmVILKEN
                                0
                                0.75
POSITIVA Y-AXELN
.....Klar
█
```

```
If X<0 och Y>0
Then
Disp "2:A KVADRANTEN"
End
```

När du är klar ska du ha åtta If-strukturer (fyra för kvadranterna och fyra för halvaxlarna).

Nedan visar vi hur programmet kan se ut.

```
Input          [Obs, ingen variabel här]
Disp X,Y
If X>0 och Y>0
Then
Disp "1:A KVADRANTEN"
End
If X=0 och Y>0
Then
Disp "POSITIVA Y-AXELN"
End
If X<0 och Y>0
Then
Disp "2:A KVADRANTEN"
End
If X<0 och Y=0
Then
Disp "NEGATIVA X-AXELN"
End
If X<0 och Y<0
Then
Disp "3:E KVADRANTEN"
End
If X=0 och Y<0
Then
Disp "NEGATIVA Y-AXELN"
End
If X>0 och Y<0
Then
Disp "4:E KVADRANTEN"
End
```