Kapitel 4 Loopar

Övning 2: While…End-loopen

I denna andra aktivitet för kapitel 4 ska du lära dig att använda **While**…**End**-loopen. Vi jämför den med **For**… -loopen and och visar varför den är mer kraftfull och mångsidig än **For…-**loopen.

Syfte:

- Lära sig strukturen hos **While…End**-loopen.
- Jämföra den med **For…End** -loopen.
- Se hur den används för att säkerställa giltiga input-värden.

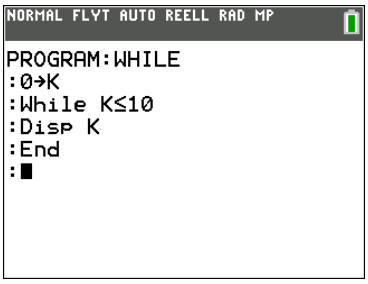
While… End-Loopen

While…End-loopen kommer att fortsätta att gå i en slinga (loop=slinga) så länge som dess <villkor> är *sant*. Det ser ut så här:

While <villkor>
  <loopkropp>
End

Obs

<villkor> är ett logiskt uttryck, t.ex. **X>0.**
<loopkroppen> är en uppsättning satser, som inkluderar andra loopar och **If***-strukturer. Den processas närhelst <villkor> är* sant*.*
End *används för att indikera slutet på <loopkroppen>. Vid* **End***-satsen loopar programmet tillbaka till* **While***-satsen och undersöker <villkor> igen.*



En oändlig loop. Varför?

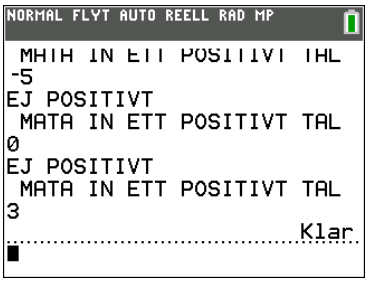
Vad åstadkommer detta program?

Ge först ett begynnelsevillkor för **While**: ange ett värde så att villkoret är ordentligt fastställt som Sant eller Falskt. Om begynnelsevillkoret är Falskt så hoppar programmet över loopen. Om villkoret är Sant så processas loopkroppen. **0→K** på första raden i programmet sätter begynnelsevillkoret till Falskt. Utan det så kan man inte veta vad som händer eftersom vilket värde som helst kan ha lagrats i variabeln **K** innan programmet körs.

Någonstans i <loopkroppen> ska det finnas en sats som påverkar <villkor> så att loopen eventuellt slutar och satserna efter loopen processas. Vanligtvis finns denna sats nära slutet av <loopkroppen>. **K+1→ K** ser till att **K** eventuellt blir större än 10.

Kontroll av giltig Input med While…End

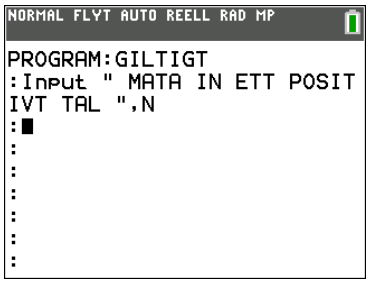
Vi ska nu skriva en del av programmet (kodsegment) som ser till att användaren matar in ett positivt tal och ger ett meddelande tillbaka om man matar in ett ogiltigt värde och att man då får uppmaningen att mata in ett nytt värde.



Utdata från kodsegmentet visas till höger, Vi har matat in ett negativt tal, 0 och sedan ett positivt tal.

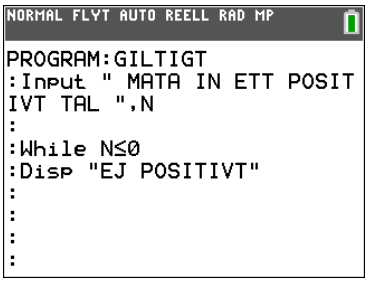
Obs: TI-84 Plus-användare med en mindre skärm måste kanske skriva kortare textsträng än ”MATA IN ETT POSITIVT TAL” eftersom antalet tecken som får plats på en rad är mindre.

Vi börjar med en **Input**-sats med ett meddelande om att få ett värde från användaren.



Sedan börjar **While**-loopen, som kontrollerar om det inmatade värdet är negativt. Om värdet är negativt så visas ett felmeddelande. Se **Disp**-satsen.

Om N>0 så skippas loopkroppen.



Till slut använder vi **Input**-satsen igen i slutet av loopkroppen för att efterfråga ett annat värde från användaren. End avslutar loopkroppen.

Detta kodsegment kommer att fråga efter ett värde från användaren och se till att inmatningen är ett positivt tal.