Kapitel 4: Loopar — Övning 2: While

I denna aktivitet kommer du att lära dig om den mest mångsidiga av looparna, nämligen While.

Syfte:

- Skriva en enkel While-loop
- Använd while-loop för att säkerställa giltig datainmatning

Om While

While…EndWhile-loopen kommer att fortsätta loopningen så länge som dess <villkor> är Sant. Det ser ut så här:

<initiera villkoret**>**

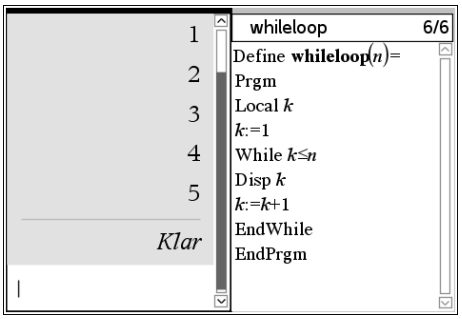
While <villkor>

<loopkropp>

EndWhile

- *Initiera* refererar till upprättandet av en eller flera variabler så att While-satsen kan korrekt utvärdera villkoret första gången. Initieringen fastställer ett värde Sant eller Falskt för variabeln. Om det första villkoret är falskt, hoppas loopen helt över. Om villkoret är Sant, då bearbetas loopkroppen.
- <villkor> är ett logiskt uttryck, t.ex. **X>0**.
- <loopkroppen> är en uppsättning av satser, inklusive andra loopar och **If**-strukturer. <loopkroppen> processas när <villkor> är Sant.
- **EndWhile** används för att indikera slutet på <loopkroppen>. Vid **EndWhile**-satsen loopar programmet tillbaka **While**-satsen och testar <villkoret> igen. Om villkoret är falskt förbigås loopen. Om villkoret är Sant så processas loopkroppen igen.

K:=1 I början av detta program ställer in begynnelsevillkoret till ett känt falskt värde. Utan denna initiering kan variabeln **K** ha vilket lagrat värde som helst.



Någonstans i <loopkroppen> måste det finnas en sats som påverkar <villkoret> så att loopen eventuellt avslutas och satser efter loopen kommer att processas. Vanligtvis är denna sats placerad i slutet av <loopkroppen>. **k:=k+1** säkerställer att k så småningom kommer att öka till att vara större än *n*.

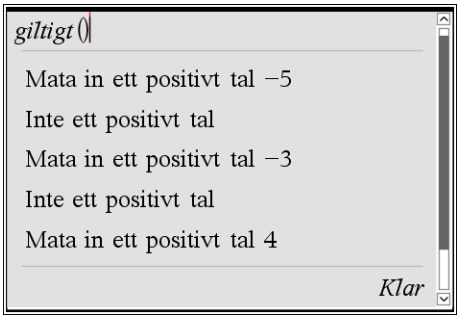
Programmet motsvarar **While**-motsvarigheten av **For**-loopen:

```
For k,1, n
    Disp k
EndFor
```

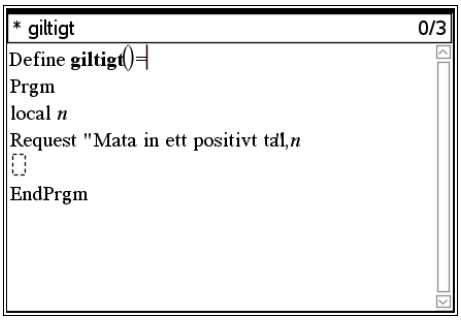
Kontroll för giltig inmatning med While…End

Vi ska nu skriva en del av ett program (ett kodsegment) som ser till att användaren matar ett positivt tal, meddelar användaren om en inmatning är ogiltig och uppmanar användaren att mata in ett annat tal.

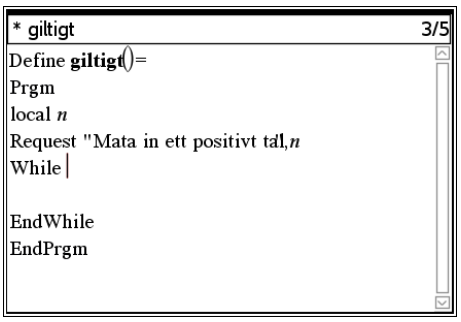
Körningen visas till höger. Vi har matat in några negativa tal och sedan ett positivt tal för att se att det fungerar



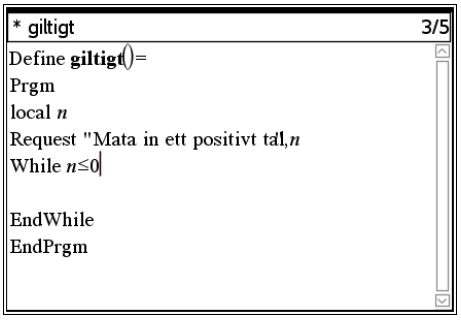
1. Vi börjar med att starta ett nytt program som vi döper till **giltigt.**
2. Vi skapar en lokal variabel ***n***, och använder en **Request**-sats för att få ett värde från användaren. Observera uppmaningen att mata in ett **positivt tal**.



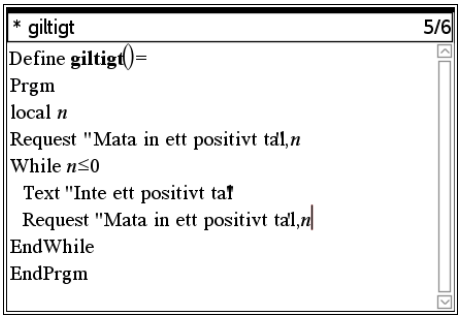
3. Infoga nu While-strukturen från Kontrollmenyn. Både While- och EndWhile-kommandona klistras in i koden och markören placeras efter ordet While.



4. Skriv nu villkoret $n \leq 0$. Symbolen ≤ finns under Beteckningar i verktygslådan. Klicka då först på 📖 och välj Beteckningar.



5. Slutför nu *loopkroppen* genom att lägga till ett felmeddelande där du använder **Text**-sats och en annan **Request**-sats som låter användaren mata in ett värde för *n* igen.



Notera nu de TVÅ **Request**-satserna:

- Den första används för att initiera villkoret ($n \leq 0$). Om ett positivt tal matas in i detta läge kommer loopen inte att processas alls
- Men om 0 eller ett negativt tal matas in kommer loopkroppen att visa ett felmeddelande och efterfråga ett annat värde.

Loopen kommer att fortsätta så länge ett icke positivt värde matas in.