Kapitel 5: Listor, grafik och dynamiska program

Övning 1: Programmering med listor

I denna aktivitet kommer du att lära dig hur man använder listor i program för att skapa intressanta spridningsdiagram.

Syfte:

- Beskriva det grundläggande om att använda **listor** i program
- Skriva program som använder **listor** för att skapa spridnings-diagram

TI-Nspire™ Basic stödjer inte direkta programsatser för grafik. Ett program kan definiera funktioner för grafritning och kan skapa listor för att producera spridningsdiagram men kan inte plotta punkter direkt. I detta kapitel ska vi närmare utveckla följande:

1. Arbeta med listor i ett program
2. Ställa in Grafappen för att visa ett spridningsdiagram utifrån listorna
3. Använda dynamiska program som kan köras på begäran.

Definiera listor

Listor definieras med data innanför klammerparenteser, { }. För att skapa en tom lista kan du använda en sats som t.ex. **minlista:={ }** utan blanksteg mellan parenteserna.

Listelementen (värden mellan parenteserna) adresseras med hakparenteser efter listnamnet, t.ex. **minlista[3]**, som refererar till det tredje elementet i **minlista**.

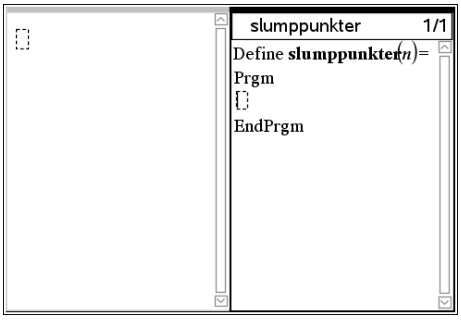
Du kan lägga till en lista genom att lagra ett värde i positionen precis efter det sista värdet i listan. Om en lista t.ex. innehåller tre element 12, 7 och 2, så kan du lägga till ett element genom att lagra ett värde i position 4. Om du skriver **minlista[4]:=17**, så resulterar det i **minlista** = {12, 7, 2, 17}. **Dim(**listnamn**)** ger information om antalet element i listan och används ofta för att lägga till ett element till en lista:

minlista[Dim(minlista)+1]:=<*något värde*>

Programmera slumpade punkter

Vi börjar nu med att skriva ett program som ger ett slumpgenererat punktmönster i appen Grafer.

1. Skapa först ett nytt program, slumppunkter, och använd argumentet *n* i programeditorn.
   - *n* representerar antalet punkter som programmet skapar.
   - Programmet skapar (eller reviderar) två ”globala” listor med slumptal som används för att plotta ett spridningsdiagram i grafappen.



Slumptalsgeneratorer

Det finns flera slumptalsfunktioner hos TI-Nspire CX. Det två vanligaste är **rand()** och **randInt( )**.

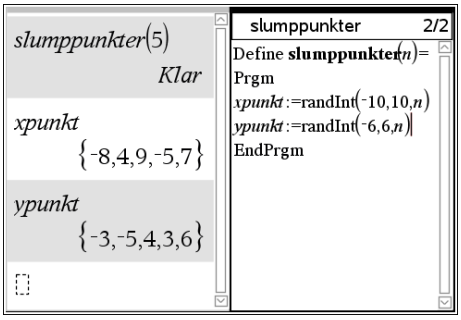
- **rand**() skapar ett slumpmässigt *decimaltal* mellan 0 och 1.
- **randInt**(*a,b*) skapar ett slumpmässigt *heltal* mellan a och b, så **randInt**(1,6) skapar ett slumptal från 1 till och med 6. Försök själv i appen Räknare.

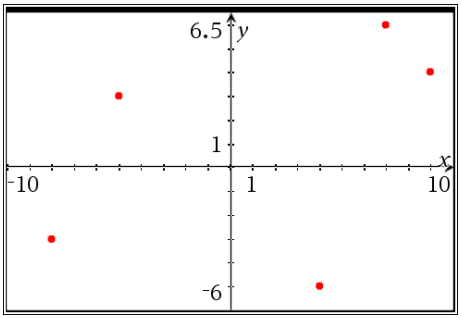
Här är vårt första försök med slumptalslistor:

xpunkt := randInt(-10, 10, *n*)
ypunkt := randInt(-6, 6, *n*)

2. Det tredje argumentet i de två uttrycken ovan ser till att **randInt** funktionen skapar en *lista* med *n* slumpmässiga heltal i ett specifikt intervall istället för ett enstaka slumptal.
3. Tryck **ctrl-B** för att lagra programmet. Kör programmet i appen Räknare med ett ett litet argument, t.ex. 5. Observera värdena hos xpunkt och ypunkt. Dina egna värden skiljer sig naturligtvis från de värden som visas i skärmbilderna till höger.
4. Lägg till en Grafapp och förbered plottning av ett spridningsdiagram för (xpunkt, ypunkt). Du väljer då Grafinmatning/Redigera från verktygslådan och sedan Spridningsdiagram i rullgardinsmenyn**.**
5. Mata in ***xpunkt*** för x-listan, och gå nedåt till y-listan, och mata in ***ypunkt***.

6 Tryck på [enter].





Intervallen för slumptalen är ungefär samma som standardinställningen för graffönstret.

När du nu har testat att programmet fungerar så gå tillbaka till Räknareappen och välj ett större men inte alltför stort värde på *n*. Kör programmet igen. Skärmbilden visar ett resultat för *n* =30.