

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.

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 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**.

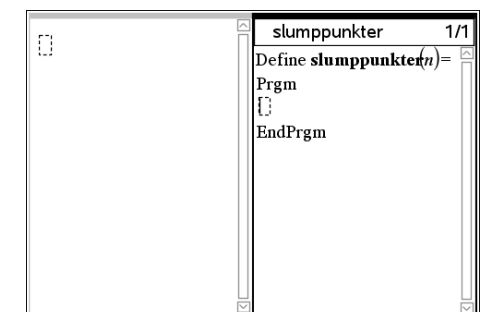
Du kan lägga till en lista genom att lagra ett värde i positionen precis efter det sista värdet i listan. Om en lista t.ex. innehåller tre element 12, 7 och 2, så kan du lägga till ett element genom att lagra ett värde i position 4. Om du skriver **minlista[4]:=17**, så resulterar det i **minlista = {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 slumpantal som används för att plotta ett spridningsdiagram i grafappen.



Slumptalsgeneratorer

Det finns flera slumptalsfunktioner hos TI-Nspire. 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.

Lärarkommentar: Datorer använder algoritmer för att alstra "pseudo-slumpmässiga" tal. Funktionen **RandSeed n** kan användas som 'frö' till slumptalsgeneratorn. Om du använder samma värde på n kommer funktionen att generera en *specifik* lista med slumptal. Slumpmässigheten kommer sig av att de tal som genereras i det långa loppet kommer att vara jämnt fördelade på valt intervall. För att göra listan slumpmässig kan du ge ditt eget slumptal till funktionen *RandSeed*. I Programeditorn kan du t.ex. använda katalogen för att hitta slumptalsfunktioner. Alla funktioner heter något som börjar med ordet rand.

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 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.

Lärarkommentar: För att se värdena för *xpunkt* och *ypunkt* så skriver du bara in namnen och trycker på enter. Du kan också klicka på knappen **var** och välja variabelnamn från listan.

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].

Lärarkommentar: När du ställer in variablerna för ett spridningsdiagram och efter det att du matat in x-variabeln är det lätt att göra misstaget att trycka på [enter] innan du matar in y-variabeln. Detta orsakar ett fel eftersom editorn "tror" att du är klar. Se alltså till att mata in båda variablerna korrekt. Om du gjort fel så trycker du bara på [tab] och försöker igen. Graffönstret kan "rensas" genom att man gömmer axlarna och tillhörande text

```

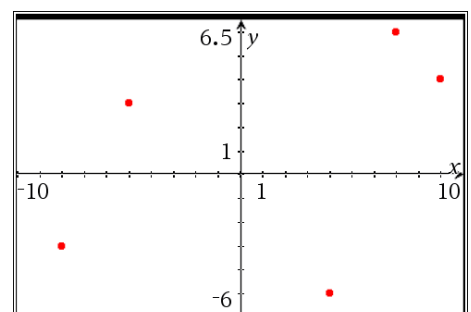
slumppunkter(5)
Klar
xpunkt
{-8,4,9,-5,7}
ypunkt
{-3,-5,4,3,6}

```

```

slumppunkter 2/2
Define slumppunkter(n)=
Prgm
xpunkt:=randInt(-10,10,n)
ypunkt:=randInt(-6,6,n)
EndPrgm

```



10 Minutes of Code

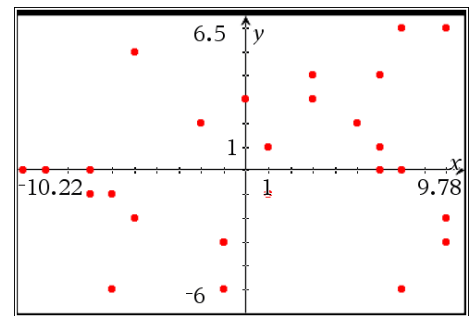
TI-Nspire-teknologi

KAPITEL 5: ÖVNING 1

LÄRARKOMMENTARER

Intervallen för slumpalen är ungefär samma som standardinställningen för graffönstret.

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



Lärarkommentarer:

- Standardfönstret är $[-10, 10] \times [-6.67, 6.67]$ så listorna fick heltalsvärden som ligger inom dessa intervall.
- De efterföljande aktiviteterna i detta kapitel tillgodoser behovet av att hoppa mellan apparna Räknare och Grafer i denna aktivitet. Allting kommer enbart att styras från grafappen.
- Det är fullt möjligt att programmet slumpmässigt alstrar samma par av heltal mer än en gång. Detta betyder att du ser färre punkter i spridningsdiagrammet än du förväntar dig. Detta är en utmärkt möjlighet att närmare utforska följande: Vilken är sannolikheten i en uppsättning av n slumpmässiga heltal i intervallet $[-10, 10]$ att två värden är lika? Vad är sannolikheten att detta kan hända två gånger (dvs. välja samma par av tal två gånger). Det finns fler frågor att ställa!