

Primtalslistor med programmering

I denna aktivitet så har vi två små program som tar fram listor på primtal inom ett visst intervall. Vi visar här programmen bredvid arandra.

```
Define primtal()=
Prgm
Local i
i:=2
While i≤100
If isPrime(i) Then
Disp i
EndIf
i:=i+1
EndWhile
EndPrgm
```

```
Define primtal2()=
Prgm
For i,2,100
If isPrime(i) Then
Disp i
EndIf
EndFor
EndPrgm
```

I programmen används s.k. loopar (slingor) och det finns tre grundläggande loopar i TI-Nspire TI-Basic: For, While, och Loop. En loopstruktur ger ett program förmåga att processa en uppsättning av satser om och om igen, antingen upprepning över en sekvens av värden (precis som i For-loopen) eller tills ett speciellt villkor är uppfyllt (eller inte) som i While och Loop. Programmen ovan visar detta på ett enkelt sätt.

Man kan läsa mer om loopar i aktiviteterna [10 Minutes of Code](#), kapitel 4 Övning 1 och Övning 2.

Kapitel 4: Loopar **Övning 2: While**

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

Syfte:

- Skrivna en enkel While-loop
- Använd while-loop för att säkerställa giltigt 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
```

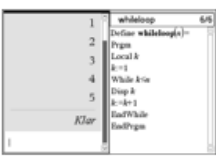
- *initiera* refererar till upprättandet av en eller flera variabler så att While-satsen kan korrekt utvärdera villkoret första gången. Initieringen fastställer ett värde Sant eller Falskt för variabeln. Om det första villkoret är falskt, hoppas loopen helt över. Om villkoret är Sant, då bearbetas loopkroppen.
- <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 looper 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 på <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
```



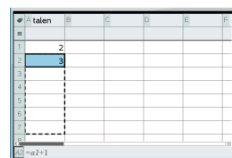
Se till att eleverna kör programmen. Man får en lista med primtalen under varandra. Man kan markera hela listan och kopiera den om man vill:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61
67 71 73 79 83 89 97

Ovan har vi klippt in dem i det ordbehandlingsprogram som det här dokumentet är skrivet i.

På de följande sidorna visar vi hur man kan arbeta med primtal i kalkylappen hos TI-Nspire. I dokumentet finns det tydliga anvisningar. Man använder den mycket användbara funktionen **"Fylla"**.

Vi ska nu visa hur man kan få en lista på de första 100 positiva heltalen och se om de är primtal. Först ska du ha en lista 1-100 i kolumn A. I cell a1 skriver du 2, som är det första primtalet. I cell a2 skriver du sedan "=a1+1" och trycker enter. Med markören i a2 högerklickar du och använder funktionen **Fylla**. Markera ner till och med rad 199 och tryck enter. Klart!



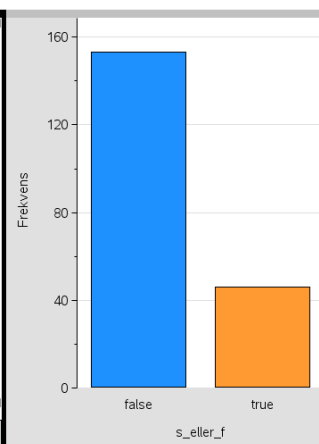
I kolumn B, som vi kan kalla faktorisering, skriver vi i b1 "=factor(a1)" och trycker enter. Med markören i b1 använder du funktionen **Fylla** på samma sätt. Se nästa sida. Rensa gärna och gör om det hela!

Vi ska nu använda funktionen **isPrime**. Gör nu på samma sätt som i kolumn A och B: skriv "=isPrime(a1)" i cell c1. Markera cell c1 och använd funktionen **Fylla** precis som tidigare.

Du får nu en lista där det står **true** eller **false**. Funktionen **testar** alltså om ett tal är ett primtal.

Till höger, där vi plottat stapeldiagram, ser vi att knappt en fjärdedel av alla tal upp till 200 är primtal. I aktiviteten "Undersöka primtal" kan du undersöka mer om primtalen. Undersök t.ex. hur stor *andel* primtal det finns bland talen 2-1000.

A	talen	B	faktorisering	C	s_eller_f
1	2			2	true
2	3			3	true
3	4	2^2			false
4	5			5	true
5	6	2*3			false
6	7			7	true
7	8	2^3			false
8	9	3^2			false
9	10	2*5			false
10	11			11	true
11	12	2^2*3			false



Vi visar här ett sätt att ta fram andelen primtal inom olika intervall. Detta finns alltså inte med i TI-Nspire-dokumentet.

kolumn D: här har vi använt formeln

$$=when(c1=true,1,0)$$

Uttrycket betyder att vi ska ersätta true med 1 i cell d1. Om c1 inte är "true" så ska det ersättas med 0.

Sedan använder vi funktionen **Fylla** för att göra motsvarande operation för alla rader.

Kolumn E: Här ska vi skapa en ackumulerad lista av data i kolumn D. I cell E1 skriver vi 1 och i cell E2 skriver vi =e1+d2. Sedan använder vi funktionen **Fylla**.

$$E2 = e1 + d2$$

Kolumn F: Här skriver vi nedanstående formel direkt i formelfältet.

$$andel_pt := \text{approx} \left(\frac{\text{ackumulerat}}{\text{talen} - 1} \cdot 100 \right)$$

Så här blir det.

	talen	faktorisering	s_eller_f	är_primtal	ackumulerat	andel_pt
=						=approx(ackumu
1	2	2	true	1	1	100.
2	3	3	true	1	2	100.
3	4	2^2	false	0	2	66.6667
4	5	5	true	1	3	75.
5	6	2*3	false	0	3	60.
6	7	7	true	1	4	66.6667
7	8	2^3	false	0	4	57.1429
8	9	3^2	false	0	4	50.
9	10	2*5	false	0	4	44.4444
10	11	11	true	1	5	50.

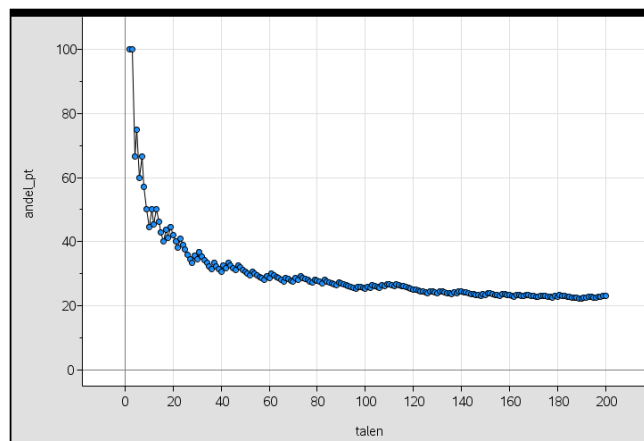
F andel_pt:=approx(ackumulerat / (talen-1) * 100)

Vi kan nu rita ett diagram som visar hur andelen primtal verkar sjunka för allt större tal intervall.

Vi visar i diagrammet andelen primtal, i procent) i intervallet 2-200. Andelen primtal ≤ 200 är ca 23 %.

Låt intresserade elever göra samma beräkning men för tall upp till 100. Eftersom talen är så snyggt ordnade i rader och kolumner kan man lätt beräkna antalet primtal.

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997	1009	1013



Diagrammet ovan är gjort med appen Data&Statistik men man kan också göra det i grafappen. Se nedan.

