Innehåll

Grafräknaren och diskret matematik

I Natur och Kulturs lärobok *Matematik 3000 diskret matematik* finns då och då en grafräknarsymbol i marginalen. Den markerar att du kanske kan behöva hjälp med hur räknaren ska användas. Här finns hjälp och tips till dessa uppgifter. Här finns förslag till hur du kan använda din räknare också till en del andra uppgifter, inte minst i avsnitten om programmering. Hjälpen är framför allt anpassad till modellerna CFX-9850GB+ och fx-9750G. De övriga grafräknarna i Casios utbud är såpass lika i uppbyggnad och logik, att du knappast ska ha några problem att följa instruktionerna som ges på hjälpsidorna, även om du t ex har en CASIO FX-2.0.

När du köpte din Casio grafräknare, ska du ha fått en cd-skiva, som heter *Upptäck Matematiken*. Den är gjord som en komplettering till din matematikbok, för att du ska få maximal glädje av din grafräknare, och för att du ska lära dig matematik så effektivt som möjligt med grafräknarens hjälp. *Upptäck Matematiken* innehåller över femtio animerade filmer om matematik och grafräknaren, som täcker in det mesta inom samtliga gymnasiets matematikkurser. Filmerna kan spelas upp på svenska eller på engelska, så skivan är också användbar om du vill lära dig ”matematik på engelska”.

Till cd-skivan finns en Aktivitetsbok med en hel del laborativa matematikuppgifter. Boken, liksom cd-skivan, kan din lärare beställa kostnadsfritt från:

Sense Office AB,
08-504 103 10.
www.sense-ab.se

Den finns också som pdf-fil, som kan hämtas från Senses hemsida eller från Broman Planetariums hemsida, www.planetarium.se. Därifrån kan du också titta på filmerna i *Upptäck Matematiken*.

På Sense hemsida finns ytterligare material speciellt för lärare. Där finns bl a ett femtiosidigt kompendium som beskriver grafräknaren i undervisningen och bilder som underlag till t ex OH-bilder av de olika räknarna. Där finns också ett kompendium som beskriver de olika versionerna av den tekniska räknaren CASIO fx-82, och hur de kan nyttjas i matematikundervisningen.

När vi beskriver hur du ska trycka på din räknare, skriver vi alltid med fet, kursiv stil, t ex: ***2 (3 + 4) EXE***. Ord inom hakparenteser, t ex ***[Solve],*** refererar till menyer, där du väljer med någon av knapparna ***F1*** ... ***F6***. Om du trycker på ***OPTN***, så får du en meny, där det sista menyvalet är en högerpil. Den anger att ”det finns mera”. Knappen ***F6*** bläddrar då mellan de sidomenyer som finns.

Vad handlar diskret matematik om?

På s 7 i läroboken finns en ruta lite allmänt om diskret matematik. En uppgift som finns där lyder:

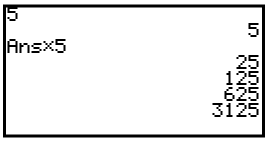
Visa att alla tal på formen $5^n - 4n - 1$ är delbara med 16.

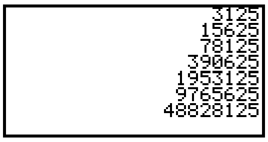
Att kunna lösa en sådan uppgift handlar ofta om att kunna finna något slags mönster. Om du skapar en tabell över tal på formen $5^n - 4n - 1$, så är ett mönster svårt att finna. Men om du tittar på tal av formen 5^n, så utkristalliseras snart ett:

Öppna RUN-fönstret och skriv:

5 EXE

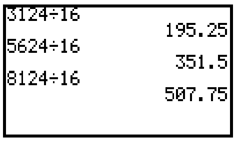
×5 EXE EXE EXE ...

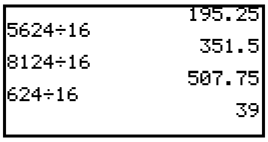




Alla tal med minst fyra siffror slutar på 3125, 5625, 8125 eller 0625, och denna svit av slutsiffror upprepas gång på gång.

Talet 10 000 är delbart med 16. (Kontrollera detta!) Om därför de sista fyra siffrorna i ett tal är delbara med 16, så är hela talet delbart med 16. Vi kontrollerar delbarheten i de fyra talen ovan minskade med ett:





Talet 3124 är inte delbart med 16, men 3120 är det. Lägg nu märke till att $3125 = 5^5$. I den ursprungliga formeln hade vi $5^5 - 4 \cdot 5 - 1 = 3124 - 4 - 16$.

5624 behöver på motsvarande sätt minskas med 8 för att bli delbart med 16, och 8124 behöver minskas med 12. Tal vars fyra sista siffror är 0624 är delbara med 16.

I nästa sekvens får vi $3124 - 4 - 32$ osv.

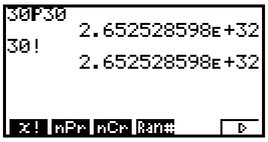
Bevisgången går att formalisera mera med hjälp av modulusoperatorn, som beskrivs längre fram i boken. Exemplet här är ett exempel på den nytta man kan ha av räknaren, när man experimenterar med ett uttryck som man vill bevisa något om.

Permutationer och kombinationer

Öppna RUN-fönstret och välj ***OPTN***. I första sidomenyn finns valet ***[PROB]***. Där finns de funktioner du behöver för att beräkna antalet permutationer och antalet kombinationer. Där finns också en slumptalsgenerator, som gör det möjligt att experimentera med sannolikheter.

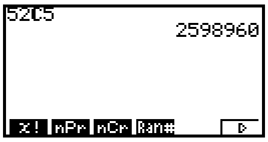
Att ur en förening med 4 medlemmar välja en ordförande, en sekreterare och en kassör kan göras på P(4, 3) olika sätt. På räknaren kan du skriva ***4 [nPr] 3 EXE***.

Antalet sätt att placera 30 elever i 30 bänkar kan bestämmas genom P(30, 30), men det är ju faktiskt detsamma som 30!. Skriv ***3 0 [x!] EXE***.



Om tre personer ur föreningen med fyra medlemmar ska åka på en konferens, så beräknas antalet kombinationer med $\binom{4}{3}$ eller C(4, 3). Motsvarande funktion på räknaren heter ***[nCr]***, och du kan skriva ***4 [nCr] 3 EXE***.

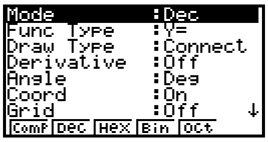
Antalet pokerhänder med 5 kort ur en kortlek med 52 kort beräknar vi också med ***[nCr]***.



Något om heltalsräkning

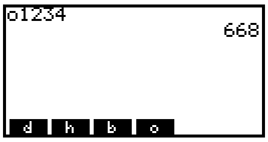
Här arbetar vi uteslutande i RUN-fönstret om inget annat sägs.

Talteori handlar om egenskaper hos framför allt heltal. Det går att ställa in räknaren för just heltalsräkning. Under ***SHIFT SET UP*** finns alla olika inställningar du kan göra på din räknare. Den översta inställningen handlar just om decimalräkning eller heltalsräkning.

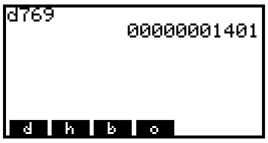


[Comp] är inställningen för vanlig räkning, medan de övriga ger heltalsräkning i baserna 10 (decimala tal), 16 (hexadesimala tal), 2 (binära tal) resp 8 (oktala tal). Vi har i bilden ställt in räknaren för räkning i basen 10.

När du gjort omställningen och tryckt på ***EXIT***, så får du en meny med valen ***[d~o]*** och ***[LOG]***. Det första menyvalet låter dig skriva in tal i en annan bas än den som just nu är aktuell. För att räkna om talet $1234_{\text{åtta}}$ till ett tal i basen tio, så skriver du ***[o] 1 2 3 4 EXE***, där alltså ***[o]*** anger att det inskrivna talet är oktalt.

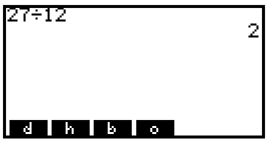


För att skriva 769_{tio} i basen åtta, så ställer du först in Mode: ***[Oct]*** under ***SHIFT SET UP***, och skriver sedan ***[d] 7 6 9 EXE***.

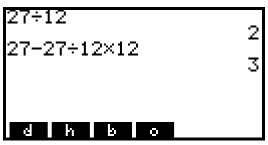


Modulusoperatorn

Heltalsdivision, m ÷ n, på räknaren ger som resultat det största heltal som är mindre än eller lika med motsvarande operation i ***[Comp]***-läge.



Egentligen är $\frac{27}{12} = 2\frac{3}{12}$, vilket vi ofta uttrycker som 2 med resten 3. Det vi egentligen beräknar, när vi beräknar 27 (mod 12) är resten i divisionen 27÷12. Det kan vi göra på räknaren med följande beräkning: ***2 7 - 2 7 ÷ 1 2 × 1 2 EXE***.

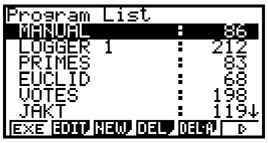


I heltalsaritmetiken på räknaren är $27-27 \div 12 \times 12 = 27 - 2 \times 12 = 27 - 24 = 3$. Visserligen har division och multiplikation lika prioritet (men högre än subtraktion), men de beräknas i den ordningen de står.

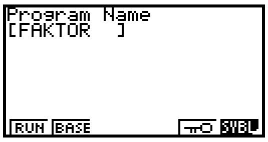
Faktorisering i primfaktorer

Som du kanske redan märkt, eller åtminstone snart kommer att märka, så har man stor nytta av att dela upp tal i primfaktorer. Handlar det om små tal, är det ganska lätt att göra i huvudet, men stora tal blir strax svårare. Är det t ex så självklart att talet 302 837 är ett primtal? Det kan vara bra att ha ett program för faktorisering på räknaren.

När du vill skriva ett program på räknaren, så börjar du med att gå in i PRGM-fönstret. Där finns en lista över de program som redan finns i räknarens programminne samt en meny.



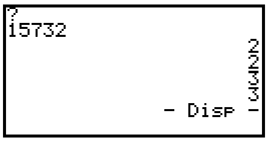
När du ska skriva ett nytt program, så trycker du på ***[NEW]***, vilket ger ett fönster där du ger ditt nya program ett namn.

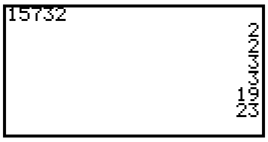


Avsluta med ***EXE***, så får du ett fönster, där du kan skriva programmet.

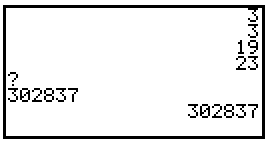
På nästa sida finns hela programmet listat och kommenterat.

När du skrivit programmet, gått ur programmeringsläget med ***EXIT***, och vill köra det, så är det enklast att markera programmet och välja ***[EXE]***. Vi provar med att faktoruppdela 15732:





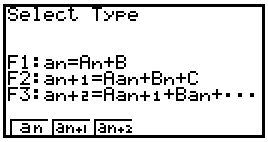
Om du direkt trycker på ***EXE***, så körs programmet en gång till, och vi kan prova med talet 302 837.



[?] → X ↵	Räknaren skriver ut ett frågetecken och väntar på ett tal, som lagras i minne X. Det är det talet som ska faktoriseras. ***[?]*** finns under ***SHIFT PRGM***.
2 → P ↵	Talet två, som är det minsta primtalet, lagras i minne P.
[While] P ≤ √X ↵	Om X inte är ett primtal, så är minsta faktorn högst √ X. Vi behöver aldrig testa längre än så. Kommandot ***[While]*** finns under ***SHIFT PRGM [COM]*** i en sidomeny. Om villkoret i satsen inte är uppfyllt, så sker hopp till raden efter ***[WhileEnd]***.
X ÷ P → Y	
[If] Y = [Int] Y ↵	Vi testar om Y är ett heltal, för i så fall är X delbart med P. I annat fall ska P få nästa udda tal som värde. ***[If]*** finns under ***SHIFT PRGM [COM]***.
[Then] P [◢]	Satsen skriver ut P och väntar tills man trycker på ***EXE***. Den s k ”stoppbocken” finns under ***SHIFT PRGM***
Y → X ↵	X får ett värde där faktorn P divideras bort. Sker detta, så hoppar programmet till ***[WhileEnd]***, varifrån hopp sker tillbaks till ***[While]***, och samma faktor testas en gång till.
[Else] [If] P = 2 ↵	Hit hoppar programmet om villkoret i ***[If]***-satsen är falsk.
[Then] 1 → P ↵	När X inte längre är delbart med 2, så ska nästa primtal vara tre. Därför ger vi P värdet 1 inför uppräkning.
[IfEnd] ↵	Hit hoppar programmet om P inte är två. I menyn under ***SHIFT PRGM [COM]*** står det ***[I-End]***.
P + 2 → P ↵	P får nästa udda värde. Om det ursprungliga värdet på X vore delbart med t ex 9, så har programmet redan givit två treor som faktorer. Alla faktorer som programmet ger är primtal.
[IfEnd] ↵	
[WhileEnd] ↵	Hopp sker tillbaks till ***[While]***-satsen. I menyn står det ***[Wend]***.
X	Om inga villkor längre är uppfyllda, så är X ett primtal, och skrivs ut som sista faktor.

Talföljder

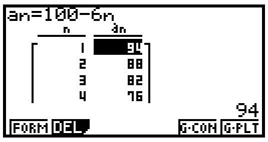
Casio-räknarnas RECUR-fönster innehåller de verktyg du behöver för att kunna studera talföljder och rekursionsformler. Välj ***[TYPE]***, så får du upp följande fönster:



a_n ger talföljder där varje tal ges av talet *n*, t ex $a_n = 100 - 6n$. Vi vill skapa en tabell med de tio första talen i talföljden. Välj därför ***[an]***. Nu kan du skriva formeln ***1 0 0 - 6 [n] EXE***.

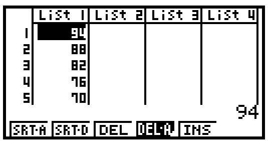
Välj nu ***[RANG]*** för att ange vilka tal i talföljden du vill se.

Tryck slutligen på ***EXIT*** och välj ***[TABL]***.



Du kan bläddra i talföljden med uppåt- och nedåtpilarna.

En beräknad tabell kan exporteras till en lista. Det går till så, att med något tal markerat i den kolumn du vill exportera, trycker på ***OPTN [LIST] [LMEM][List1]***. Om du växlar till STAT- eller LIST-fönstret, så finns talföljden där.



Talföljder som ges av rekursionsformler

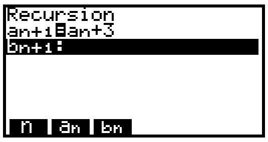
Öppna RECUR-fönstret och välj ***[TYPE]***. Här anger ***[an+1]*** rekursionsformler där varje tal i talföljden kan beräknas ur det föregående talet. ***[an+2]*** anger rekursionsformler där varje tal ges av de två föregående talen. Vi tittar på exemplen på s 60.

Exempel 1.

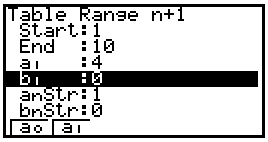
$f(1) = 4$, $f(n+1) = f(n) + 3$.

Talföljden kan också beskrivas lite mer ”räknarmässigt”: $a_{n+1} = a_n + 3;\ a_1 = 4$.

Välj ***[TYPE] [an+1]*** och skriv in formeln. Observera att ***[an]*** finns under ***[nan]***.



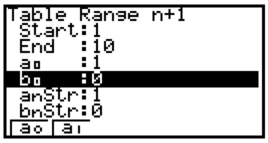
Startvärde och omfång ställs in under ***[RANG]***. Observera att startvärdet i det här fallet ska vara av typen ***[a1]***.



Tryck slutligen på ***EXIT*** och ***[TABL]***.

Exempel 2.

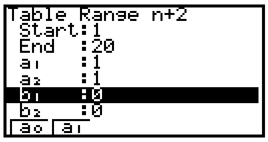
$n!$ kan definieras rekursivt med formeln $a_{n+1} = (n+1) \cdot a_n$; $a_0 = 1$. Formeln och inställning under ***[RANG]*** ser då ut så här:



I det här fallet använder vi startvärde ***[ao]***.

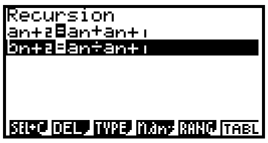
Fibonaccital och andra liknande tal

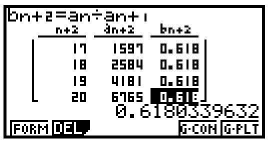
Talföljden 1, 1, 2, 3, 5, 8, 13, ..., där varje tal ges av summan av de två föregående talen, kallas Fibonaccis talföljd efter den italienska 1200-talsmatematikern Fibonacci eller Leonardo från Pisa. Talföljden ges av rekursionsformeln $a_{n+2} = a_n + a_{n+1}$; $a_1 = a_2 = 1$. Välj ***[TYPE] [an+2]***, så kan du skriva formeln och range så här:



Tryck på ***EXIT*** och välj ***[TABL],*** så får du en tabell över de första 20 Fibonaccitalen.

Vi låter nu talföljden $b_{n+2} = a_n \div a_{n+1}$, dvs kvoten mellan två närliggande Fibonaccital.





Vi kan nu observera att skillnaden mellan talen i talföljden b_n minskar för allt större n, och man kan visa att talföljden går mot det tal som kallas för det gyllene snittet.

Undersök talföljden b_n för andra startvärden än $a_1 = 1$ och $a_2 = 1$, gärna olika, gärna negativa och gärna irrationella som π eller sin(43°). Vad upptäcker du?

Man kan visa att om m och n är två godtyckliga positiva heltal sådana att $m > n$, så utgör $a = m^2 - n^2$, $b = 2mn$ och $c = m^2 + n^2$ tre heltal sådana att $a^2 + b^2 = c^2$. Det är alltså en generator för rätvinkliga trianglar med heltalssidor, s k Pytagoreiska taltripplar.

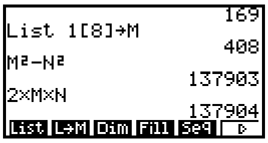
Om man väljer *n* och *m* som två på varandra följande tal i talföljden $a_{n+2} = a_n + 2a_{n+1}$, så kommer motsvarande tal *a* och *b* att vara två närliggande tal, dvs skillnaden mellan *a* och *b* är 1.

Om du vill undersöka denna egenskap närmare, så är det praktiskt att exportera tabellen över talföljden till LIST-fönstret. Då går det hyggligt enkelt att räkna med talen i talföljden även i RUN-fönstret. Vi visar som exempel där det sjunde och åttonde talen i talföljden används.

Först sparar vi det sjunde talet i minne N och det åttonde talet i minne M.

Hakparenteserna finns ovanför tangenterna ”+” och ”-”.

Nu beräknar vi lätt $m^2 - n^2$ och $2mn$.



Två närliggande heltal som kan utgöra mätetalen till kateterna i en rätvinklig triangel med heltalssidor kallas för Pytagoreiska tvillingar. Man kan visa att alla tal genererade med rekursionsformeln ovan i sin tur genererar Pythagoreiska tvillingar. Bevismetoden är ett s k induktionsbevis, den bevismetod som beskrivs på sidorna 65–67:

Visa att $n = 1$ och $m = 2$ genererar ”triangeln” 3, 4, 5.

Antag att $\left| m^2 - n^2 - 2mn \right| = 1$ (induktionsantagande).

Visa att $\left| (n + 2m)^2 - m^2 - 2(n + 2m) \cdot m \right| = 1$.

Summor

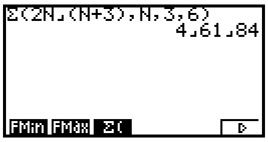
Exempel 2226.

Beräkna exakt $\sum_{k=1}^{5} k^4$ och $\sum_{n=3}^{6} \frac{2n}{n+3}$.

Räknaren har en summeringsfunktion, som finns i en sidomeny under ***OPTN [CALC]***. För att beräkna $\sum_{k=1}^{5} k^4$, kan du skriva:

[Σ( ] ALPHA K ^ 4 , ALPHA K , 1 , 5) EXE

För att få summan $\sum_{n=3}^{6} \frac{2n}{n+3}$ i bråkform, måste du mata in formeln som ett bråk.



Resultatet är $4\frac{61}{84}$.

Algoritmer och programmering

I fotnoten s 68 påpekas att man kan välja några uppgifter och programmera dem på en dator. Men CASIO grafräknare har ett BASIC-liknande programmeringsspråk, som väl lämpar sig till uppgifterna i läroboken. Vi visar med några exempel hur språket fungerar.

Uppgift 2401

Ett tal ska multipliceras med 6, subtraheras med 12, divideras med 3 och adderas med 4.

När du gått in i PRGM-fönstret, så välj ***[NEW]*** för att skapa ett nytt program. I nästa fönster ska du ge programmet ett namn, t ex EX241. Du skriver med bokstäverna ovanför knapparna, men innan du skriver siffrorna i namnet, så tryck på ***ALPHA***. Avsluta med ***EXE***.

Här följer en tabell, där vänsterkolumnen ger en listning av programmet, och högerkolumnen ger kommentarer.

"SKRIV ETT TAL" ? →X ↵	Citationstecknen gör att texten skrivs ut på skärmen. När du skriver texten, så tryck först på ***SHIFT ALPHA***. Då finns citationstecknet som val på ***F2***. ***? →X*** innebär att räknaren väntar på att ett tal följt av ***EXE*** skrivs in. Det talet lagras i minne X. Retursymbolen ↵ får du genom att trycka på ***EXE***.
X × 6 →X ↵	Här görs den första räkneoperationen. Resultatet av operationen sparas i minne X, men skriv inte ut på skärmen.
X - 1 2 →X ↵	
X ÷ 3 →X ↵	
X + 4 →X	Resultatet av den sista beräkningen skrivs alltid ut på skärmen.

Avsluta programmeringen genom att trycka på ***EXIT*** kanske ett par gånger, så att du får det första PRGM-fönstret på skärmen.

När du kör ett program, så sker det alltid i RUN-fönstret. Men det är enklast att starta det genom att markera det i PRGM-fönstret och trycka på ***EXE*** eller att välja ***[EXE]*** i menyn.

Uppgift 2404, logiska val

Programmet ska ta in vikt V och längd L på en person och beräkna Body Mass Index, BMI enligt formeln $BMI = V / L^2$. Vikten anges i kg och längden i m. Vi ska dock göra programmet "förlåtande" i den meningen att om längden matas in i cm, så ska programmet räkna om till m. Om $BMI > 25$, så ska programmet svara "OEVERVIKTIG", annars, om $BMI < 18$ ska programmet svara "UNDERVIKTIG", annars "OK".

Logiska val åstadkommes på räknaren under ***SHIFT PRGM [COM]*** med kommandona ***[If]***, ***[Then]***, ***[Else]*** och ***[I-End]***. Det sista kommandot skrivs ut som ***IfEnd*** på skärmen, och avslutar den del av ett program där logiska val görs.

Program BMI

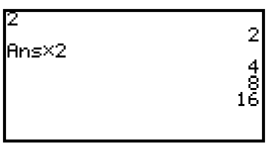
”VIKT” ? → V ↵	Lägger vikten i minne V
”LAENGD” ? → L ↵	Lägger längden i minne L
If L > 10 ↵	Här kollas om längden är inmatad i meter eller i cm; ingen människa är längre än tio meter. Olikhetstecknen finns under ***SHIFT PRGM [REL]*** (som finns i en sidomeny).
Then L ÷ 1 0 0 → L ↵	L räknas om till meter.
IfEnd ↵	Slut på hela If-satsen.
V ÷ L $\boldsymbol{x^2}$ ***→ B ↵***	BMI beräknas och lagras i minne B
If B > 2 5 ↵	Detta är en ny IF-sats, där vi undersöker BMI.
Then ”OERVIKTIG” ↵	
Else If B < 1 8 ↵	Vi nästlar en IF-sats inuti den If-sats vi skriver.
Then ”UNDERVIKTIG” ↵	
Else ”OK” ↵	
IfEnd ↵	Detta är slutpunkten för den nästlade if-satsen.
IfEnd	Slut på if-satsen som börjar på den sjunde raden, också slut på programmet.

Om du har kört programmet en gång och fått ett resultat, så kan du köra programmet en gång till genom att trycka på ***EXE***. Om du i stället trycker på ***AC*** eller utför någon annan beräkning, så avslutas programmet.

Hur reagerar programmet om Tina väger 50 kg och är 174 cm lång?

Upprepningar

Öppna RUN-fönstret och skriv ***2 EXE × 2 EXE EXE EXE***.



Varje gång du trycker på ***EXE***, så multipliceras det sist uträknade resultatet med två. Samma beräkning, men med nya ingångsvärden, upprepas gång på gång. I nästan all programmering förekommer upprepningar, antingen ett givet antal gånger eller, vilket är vanligare, tills dess att ett givet villkor är uppfyllt. Titta på sidomenyerna under ***SHIFT PRGM [COM]***. Det går bra att göra i RUN-fönstret också.

[For] [To] [Step] [Next] är den svit av kommandon som används när man vill göra något ett givet antal gånger. Typiskt ser det ut så här. Vi väljer uppgiften 2409:

Program EX2409

10 →X ↵	Lägg märke till att boken, liksom många programmeringsspråk, skriver *x*:=10. Räknaren använder tilldelningspilen i stället.
[For] 1 →N [To] 4 ↵	Här är minnet N en räknare, som första gången antar värdet 1, andra gången värdet 2 osv. Hade jag velat använda N i mina beräkningar, och att N bara skulle anta jämna värden, så hade jag kunnat skriva: ***[For] 2 →N [To] 8 [Step] 2 ↵***
X + 2 →X ↵	Här görs den upprepade beräkningen.
[Next]	Här sker hopp tillbaks till andra raden så länge N < 4. Annars avslutas programmet, och det sist uträknade resultatet skrivs ut.

[While] [WEnd] [Do] [Lp-W] är två typer av upprepningar. De ser ut så här:

Program EX2410A

5 →X ↵	
[While] X < 100 ↵	***[While]*** står alltid tillsammans med ett villkor, som också testas i denna rad. Är villkoret *inte* uppfyllt sker hopp till raden efter ***[WEnd]***, eller, som i det här fallet, programmet avslutas och resultatet av den sista uträkningen skrivs ut. Olikhetstecknet finns under ***SHIFT PRGM [REL]*** (som finns i en sidomeny).
X + 1 →X ↵	
2 X →X ↵	
[WEnd]	På skärmen står det ***WhileEnd***, och markerar slutet på en ”While-loop”.

Program EX2410B

1 →X ↵	
[Do] ↵	Så länge villkoret i rad 5 *är* uppfyllt, så sker hopp tillbaks till denna rad.

3 X →X ↵	
X + 5 →X ◢	"Stoppbocken", som finns under ***SHIFT PRGM***, anger att det beräknade värdet ska skrivas ut. Programmet väntar sedan tills dess att du trycker på ***EXE***.
[Lp-W] X ≤ 9 0	Så länge villkoret X ≤ 90 så sker hopp tillbaks till raden ***[Do]***. Annars avslutas programmet, och resultatet av den sista uträkningen skrivs ut (en gång till).