Nollställen och vertex med programmering.

I denna aktivitet ska eleverna med ett program bestämma nollställen och vertex för andragradsfunktioner som skrivs på formen $a \cdot x^2 + b \cdot x + c$. Eleverna bör ha förkunskaper om hur man löser andragradsekvationer med kvadratkomplettering och ha en viss vana att använda formeln direkt.

På de tre första sidorna ger vi en utförlig förklaring till själva matematiken i aktiviteten.

Nollställen och vertex

I denna aktivitet ska vi med ett program bestämma nollställen och vertex för andragradsfunktioner som skrivs på formen $a\cdot x^2+b\cdot x+c$. Du bör ha förkunskaper om hur man löser andragradsekvationer med *kvadratkomplettering*. Vi går inte igenom det här. Om vi löser andragradsekvationen $a\cdot x^2+b\cdot x+c=0$ (eller beräknar nollställena) får vi med TI–Nspire CAS

$$\text{solve}(a\cdot x^2+b\cdot x+c=0,x) \blacktriangleright x=\frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a} \text{ or } x=\frac{-(\sqrt{b^2-4\cdot a\cdot c}+b)}{2\cdot a}$$

För att komma fram till lösningsformlerna så använder man alltså kvadratkomplettering. Bra beskrivningar finns i de flesta läroböcker.

Uttrycket inom parentes, $\sqrt{b^2-4\cdot a\cdot c}$, kallas *diskriminant*.

Man kan skriva om uttrycket $a\cdot x^2+b\cdot x+c$ på formen $a\cdot(x-h)^2+k$. Den senare formen kallas *vertexform* eftersom *h* och *k* är *x*– och *y*–koordinaterna för vertex. Fundera på varför det är så? Även här används kvadratkomplettering

$$\text{completeSquare}(a\cdot x^2+b\cdot x+c,x) \blacktriangleright a\cdot\left(x+\frac{b}{2\cdot a}\right)^2+\frac{4\cdot a\cdot c-b^2}{4\cdot a}$$

som också kan skrivas om med negativt tecken framför *b*–termen i parentesen. Även uttrycket utanför parentesen kan förenklas. Tryckning på enter och vi får direkt tillbaka funktionen i standardform.

$$a\cdot\left(x-\frac{-b}{2\cdot a}\right)^2+c-\frac{b^2}{4\cdot a} \blacktriangleright a\cdot x^2+b\cdot x+c$$ Enligt vertexformen ovan är

alltså x–koordinaten för vertex $\frac{-b}{2\cdot a}$ och y–koordinaten är $c-\frac{b^2}{4\cdot a}$.

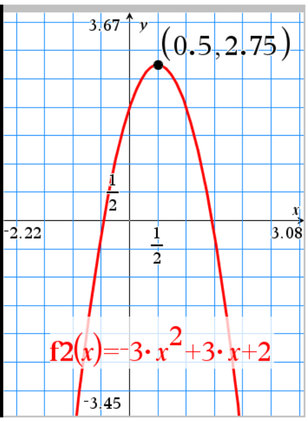
Förklara för eleverna hur man ska tolka vertexformen för en andragradsfunktion. Skilj då på funktionerna som har maximi- eller minimivärden. a<0 ger maximivärde och a>0 ger minimivärde.

Vi tar nu ett exempel med tal istället för symboler.

Vi har funktionen $-3x^2+3x+2$ som vi först skriver om som $-3(x^2-x)+2$

Det som står i parentesen ska nu vara en del av en kvadrat på formen $(x-h)^2$ och då måste det bli så här: $-3\left(x^2-x+\frac{1}{4}\right)+2+\frac{3}{4}=$

$-3\left(x-\frac{1}{2}\right)^2+\frac{11}{4}$



På sid 4 ger vi förklaring till villkorssatserna. Vi använder här ElseIf istället för flera If Then… EndIf-satser.

Förklaringar till programmet:

Vi har *tre* olika villkor i programmet och If–satsen kan då utökas

1): If **diskr**>0 Then … (då finns två reella rötter)
ElseIf **Diskr**=0 Then (dubbelrot)
ElseIf **Diskr**<0 Then (inga reella rötter)
EndIf

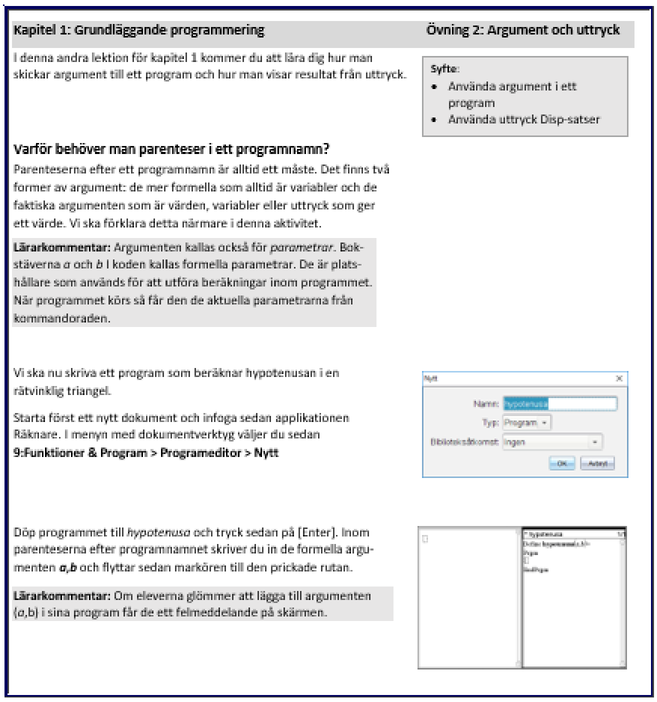
Man kunde också här ha gjort en egen If–sats för varje villkor.

I övrigt är det mest Disp–satser för sådant som ska visas på skärmen vid programkörning och sådant som samtidigt ska beräknas. Programmet är mycket utförligt och vi har använt riktiga namn för många variabler. I början av koden har vi lagt in

funk$:=a\cdot x^2+b\cdot x+c$ och **symmetrilinje**$:=x=\frac{-b}{2\cdot a}$ för att kunna plotta funktionen och symmetrilinjen. Se sid 7.

Argumenten på kommandoraden kallas för *parametrar*. Bokstäverna *a*, *b* och c i koden kallas formella parametrar. De är platshållare som används för att utföra beräkningar inom programmet. När programmet körs så får det de aktuella parametrarna från kommandoraden.

I kapitel 1, Övning 2 hos [10 Minutes of Code]-aktiviteterna så tar vi närmare upp hur man kan arbeta med argument.



Kapitel 1: Grundläggande programmering — **Övning 2: Argument och uttryck**

I denna andra lektion för kapitel 1 kommer du att lära dig hur man skickar argument till ett program och hur man visar resultat från uttryck.

Syfte:
- Använda argument i ett program
- Använda uttryck Disp-satser

Varför behöver man parenteser i ett programnamn?

Parenteserna efter ett programnamn är alltid ett måste. Det finns två former av argument: de mer formella som alltid är variabler och de faktiska argumenten som är värden, variabler eller uttryck som ger ett värde. Vi ska förklara detta närmare i denna aktivitet.

Lärarkommentar: Argumenten kallas också för *parametrar*. Bokstäverna *a* och *b* I koden kallas formella parametrar. De är platshållare som används för att utföra beräkningar inom programmet. När programmet körs så får den de aktuella parametrarna från kommandoraden.

Vi ska nu skriva ett program som beräknar hypotenusan i en rätvinklig triangel.

Starta först ett nytt dokument och infoga sedan applikationen Räknare. I menyn med dokumentverktyg väljer du sedan **9:Funktioner & Program > Programeditor > Nytt**

Döp programmet till *hypotenusa* och tryck sedan på [Enter]. Inom parenteserna efter programnamnet skriver du in de formella argumenten ***a*,*b*** och flyttar sedan markören till den prickade rutan.

Lärarkommentar: Om eleverna glömmer att lägga till argumenten (*a*,b) i sina program får de ett felmeddelande på skärmen.

På nästa sida har vi listat det ganska långa programmet. Här används alltså koefficienterna i andragradsfunktionen som argument i programnamnet.

Programkörningen ger ju ganska mycket information så det finns många Disp-satser.

Define **noll_och_maxmin**(a,b,c)=
Prgm
© vi söker här nollställen och vertex till andragradsfunktioner
$funk:=a\cdot x^2+b\cdot x+c$
$symmetrilinje:=x=\frac{-b}{2\cdot a}$
Disp "x−koordinat för funktionens vertex är ",$\frac{-b}{2\cdot a}$
Disp "y−koordinat för funktionens vertex är ",$c-\frac{b^2}{4\cdot a}$
$diskr:=b^2-4\cdot a\cdot c$
If $diskr>0$ Then
$nollställe1:=\frac{-b-\sqrt{diskr}}{2\cdot a}$
$nollställe2:=\frac{-b+\sqrt{diskr}}{2\cdot a}$
Disp "Diskriminanten är ",*diskr*
Disp "Första nollstället är",*nollställe1*
Disp "Andra nollstället är",*nollställe2*
ElseIf $diskr=0$ Then
$dubbelrot:=\frac{-b}{2\cdot a}$
Disp "Diskriminanten är ",*diskr*
Disp "dubbelroten är",*dubbelrot*
ElseIf $diskr<0$ Then
Disp "Diskriminanten är ",*diskr*
Disp "Funktionen saknar nollställen"
EndIf
EndPrgm

Här visar vi nu några programkörningar

noll_och_maxmin(1,5,6)

x−koordinat för funktionens vertex är $\frac{-5}{2}$
y−koordinat för funktionens vertex är $\frac{-1}{4}$
Diskriminanten är 1
Första nollstället är -3
Andra nollstället är -2

Klar

noll_och_maxmin(1,2,1)

x−koordinat för funktionens vertex är -1
y−koordinat för funktionens vertex är 0
Diskriminanten är 0
dubbelroten är -1

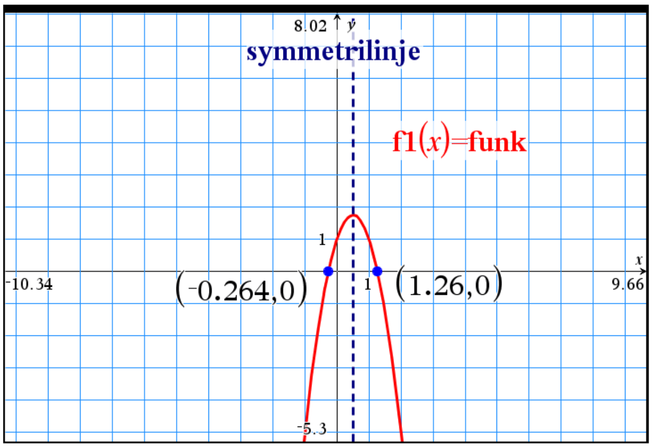
Klar

noll_och_maxmin (1,1,

x−koordinat för funktionens vertex är $\frac{-1}{2}$
y−koordinat för funktionens vertex är $\frac{3}{4}$
Diskriminanten är -3
Funktionen saknar nollställen

Klar

Så här ser grafsidan ut för funktionen $-3\cdot x^2+3\cdot x+1$. Symmetrilinjens ekvation är *x*= ½. Det är ingen funktion!



Man kan plotta sådana här lodräta linjer genom att mata in *x*=[värde] i inmatningsfönstret för relationer.

rel1(x,y)

$x=1/2$