

Reskassa

Två syskon Tina och Tim letar efter sommarjobb. De bestämmer att slå samman sina två löner från sommarjobbet för att finansiera en resa som planeras i slutet av sommaren. De vet ännu inte vilka löner de kan få för sommarjobbet. För att genomföra sitt sommarprojekt måste de samla in minst 15 000 kr. De kan inte arbeta mer än 15 dagar och det vet också att Tim, som är yngst, inte får arbeta mer än 6 timmar per dag medan Tina får jobba 8 timmar. Samtidigt har de också, beroende på ålder, olika timlön. Tim får 80 % av Tinas lön.

Vilken minsta timlön måste Tina få för att de ska få ihop tillräckligt med pengar för drömresan? Använd först grafräknaren utan att skriva något program. Du ska först plotta två olika funktioner:

- Den totala dagslönen som funktion av Tinas timlön om de jobbar maximalt antal timmar.
- Den totala inkomsten om de jobbar 15 dagar (maximal antalet dagar de kan jobba) som funktion av Tinas timlön.

Lös uppgiften genom att läsa av i plottningarna och använda räknarens grafiska verktyg.

Skriv sedan ett program i Pythonappen där syskonen kan mata in Tinas lön och det belopp de vill använda för resan. Beloppet måste vara minst 15 000 kr. Vid körningen ska sedan det antal dagar de måste jobba visas.

Vi antar att Tina får en timlön på x kronor. Det betyder att den sammanlagda lönen för en dags arbete blir

$$x \cdot 8 + x \cdot 0,8 \cdot 6 \text{ kr}$$

Vi plottar nu detta som en funktion. Vi låter x -värdena (timlönen) variera mellan 0 och 150 kr. Genom att använda trace-funktionen ser vi att vid en timlön på 90 kr för Tina så blir dagsförtjänsten 1152 kr.

Om vi antar att de jobbar i 15 dagar, som är max, så blir förtjänsten $15 \cdot (x \cdot 8 + x \cdot 0,8 \cdot 6)$

Vi plottar nu denna funktion i ett nytt fönster. Om lönen för Tina är ca 78 kr så kommer de upp i ca 15 000 kr. *Algebraiskt* så kan vi lösa detta exakt genom att ställa upp *ekvationen*

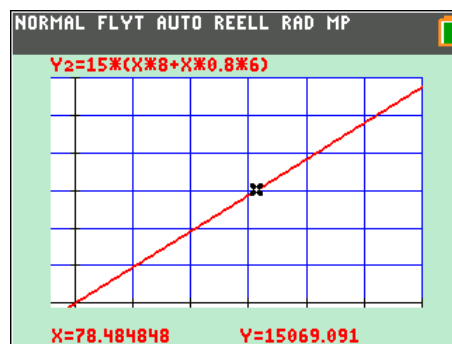
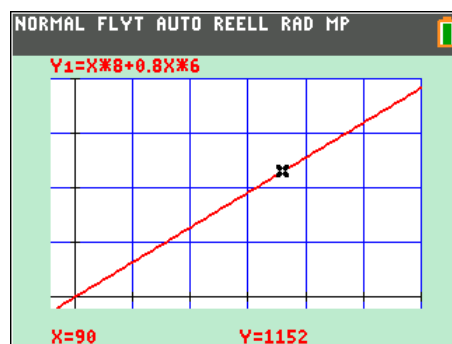
$$15 \cdot (x \cdot 8 + x \cdot 0,8 \cdot 6) = 15000 \quad \text{vi förenklar först}$$

$$120x + 72x = 15000$$

$$192x = 15000$$

Syfte med aktiviteten:

- Tolka problemtexten och visa algebraiskt och grafiskt, utanför Pythonappen, hur man kan lösa problemet
- Skriva ett program i Python där man
 - definierar en funktion och använder ett villkorsuttryck
 - Använder importkommandon från `math` import



$$x = 15000 / 192 = 78.125$$

Tina behöver alltså tjäna ca 78 kr i timmen för att de ska ha en chans att göra en resa.

Vi har här skrivit ett kort program med Pythonappen. Det heter *Olikhet*. Kör programmet med olika värden på a och c. tryck på Run (f4-tangenten) och sedan på tangenten `vars` och sedan på `enter`.

```

EDITOR: OLIKHET
PROGRAM LINE 0001
from math import *
# a är timlön för Tina
# c är reskostnad
def olik(a,c):
    if a>=78.125:
        b=a*8+0.8*a*6
        d=c/b
        svar="x= "
        return svar,ceil(d)
    else:
        return "stanna hemma"

```

Skriv sedan i värden inom parenteserna och tryck på `enter` igen.

Se skärmen till höger. Vi har här fått värdet 14 när vi matar in (95, 16000). Det som räknas ut är värdet på d "med ceil framför". d är ju c/b, dvs $16000 / (95 \cdot 8 + 0,8 \cdot 95 \cdot 6)$ och det blir ca 13,16. ceil betyder närmaste heltal uppåt och det blir 14. De kan ju inte jobba 13,16 dagar utan det är hela dagar som gäller.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running OLIKHET
>>> from OLIKHET import *
>>> olik(95,16000)
('x= ', 14)
>>> |

```

Om du nu skriver in 95, 20000 i funktionen så får du resultatet enligt skärmen till höger. 17 är ju *inte* tillåtet eftersom de kan maximalt jobba i 15 dagar. Uppgiften nu är att bygga ut programmet så att resultat som blir större än 15 blir ogiltiga.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running OLIKHET
>>> from OLIKHET import *
>>> olik(95,20000)
('x= ', 17)
>>> |

```

Kommentarer till programmet

Precis som för alla versioner av Python, måste du inkludera importkommandon, såsom "from math import *" för att använda funktioner, metoder eller konstanter som finns i math-modulen. För att till exempel utföra funktionerna ceil() måste du använda import för att importera math-modulen för användning. Till höger visas hela modulinhållet för math import. För att vara exakt så returnerar ceil(x) returnerar det minsta heltalet större än eller lika med x

I skriptet har vi en if sats som testar om ett villkor stämmer. Man kan sedan bygga ut med en eller flera elif-satser som testar ett nytt villkor om inte det första stämmer. Man avslutar if satsen med ett else om inget av de tidigare villkoren stämmer.

```

PYTHON SHELL
>>> import math
>>> dir(math)
['__name__', 'e', 'pi', 'sqrt', 'pow', 'exp', 'log', 'cos', 'sin', 'tan', 'acos', 'asin', 'atan', 'atan2', 'ceil', 'copysign', 'fabs', 'floor', 'fmod', 'frexp', 'ldexp', 'modf', 'isfinite', 'isinf', 'isnan', 'trunc', 'radians', 'degrees']
>>> |

```

På nästa sida finns hänvisningar till övningar där Pythonfunktioner som finns med i programmet Olikhet tas upp.

Grundläggande instruktioner för Pythonprogrammering tas upp i ett antal övningar med namnet *Koda med TI* eller *10 minutes of Code*. Du finner dem här på [denna länk](#).

- Funktioner (def) tas upp i kapitel 1, övning 3 och tillämpning
- Villkorssatser tas upp i kapitel 2, övning 1 och tillämpning

Du kan också läsa mer i programmeringsguiden. Den kan laddas ner från [denna sida](#).



Koda med TI: TI-84 Plus Python Edition

Korta övningar som lär dig grunderna i programmering med stöd av grafräknaren TI-84 Plus Python Edition.

Välj bland olika övningar och lär dig i steg för steg grunderna i programmering med Python.

- ✓ Kapitel 1: Starta programmering i Python
- ✓ Kapitel 2: Starta programmering på riktigt
- ✓ Kapitel 3: Starta programmering på riktigt