

Arbeta med stora tal

Först lite teori innan vi kommer in på programmeringen:

Tänk dig att totalt har glömt PIN-koden till ditt bankkort. Du vet dock att det är siffrorna 2,3,6, och 7 som förekommer i koden. På hur många sätt kan du skriva PIN-koden?

Du kanske tänker så här: Nu ska jag skriva upp alla ordningar av dessa fyra tal. Du skriver dem från det minsta till det största talet:

2367, 2376, 2637, 2673, 2736, 2763
3267, 3276, 3627, 3672, 3726, 3762
6237, 6273, 6327, 6372, 6723, 6732
7236, 7263, 7326, 7362, 7623, 7632

Det blir 24 olika uppräknings. Finns det inget enklare sätt att beräkna ordningar (kallas också *permutationer*). Jo, den första siffran kan ju man ju välja på fyra olika sätt och det andra siffran väljer man på tre sätt eftersom en siffra är upptagen. Sedan fortsätter man på samma sätt med den tredje och fjärde siffran. Det blir då

$4 \cdot 3 \cdot 2 \cdot 1 = 24$ olika sätt.

Har vi siffrorna 0 till och med 9 så blir det

$10 \cdot 9 \cdot 8 \cdot \dots \cdot 3 \cdot 2 \cdot 1 = 3\,628\,800$ sätt

Nu finns det ett sätt att skriva detta som kallas *fakultet* (*factorial* på engelska). Man skriver så här:

$4! = 24$ och $10! = 3\,628\,800$

På din räknare kan du skriva 10 och sedan trycka på tangenten $\boxed{\text{math}}$ och sedan välja SAN (eller PROB om du engelsk språkställning) och sedan alternativ 4, som är ett utropstecken.

Pröva nu att beräkna fakulteten för ett lite större tal och se vad du får. Pröva allt större tal och se hur stora tal du kan beräkna fakulteten för!

Fördjupning: n-fakulteter växer oerhört snabbt när n blir allt större.

Mata in $y1=X!$ på räknaren och visa sedan en tabell.

Fakultetsbegreppet används i en intressant del av matematiken som kallas *kombinatorik* och kombinatorik är ett moment i kurs 5. Exempel på problem som kan lösas med kombinatorik är *hur många olika femkortshänder kan du få från en kortlek med 52 kort* eller *vad är sannolikheten att du får 7 rätt i Lotto*. Be din lärare att visa några exempel men tänk på att räknaren inte kan representera hur stora tal som helst.



```
NORMAL FLYT AUTO REELL RAD MP
MA NUM KPX SAN BRÄK
1:slumP
2:nPr
3:nKr
4:!
5:slumPHeI(
6:slumPNorm(
7:slumPBin(
8:slumPHeIEjRep(
```

```
NORMAL FLYT AUTO REELL RAD MP
10!
..... 3628800
```



Vi ska nu visa ett enkelt program som beräknar fakulteten. Här finns en if-else sats och en inskjuten while-loop. Med en while-satsen upprepar du ett kodblock så länge ett villkor är sant.

while <villkor>:

indenterad kod som upprepas

I den kod som upprepas måste villkoret ändras så att det förr eller senare inte längre är sant, annars får du en oändlig loop.

Pröva nu att köra detta lilla program för några olika värden på talet n. Skärmen till höger visar en resultatet av när vi matar in talet 50. Vi får nästan precis två fyllda rader med siffror. Svaret blir ungefär $3,04 \cdot 10^{64}$.

Dra till med något riktigt stort tal. Arbeta i steg och testa hur stort tal du kan mata in, och få ett korrekt resultat på beräkningen.

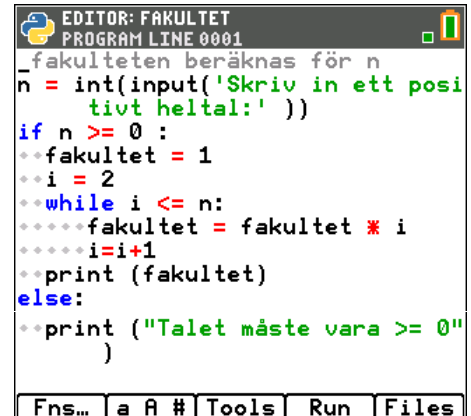
För att få storleksordningen av resultaten lite lättare att läsa av så kan man ändra i printsatsen så att man får exponentiellt format eller det vi kallar grundpotensform på svenska.

Vi har här skrivit:

print("{:.3e}".format(fakultet))

3 står här för antalet decimaler. Strängformatet .format() finns under I/O.

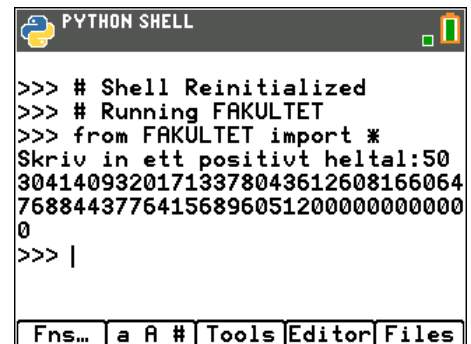
100! blir då t.ex. $9.333e+157$. Ett ruggigt stort tal. Antalet atomer i universum uppskattas till ungefär 10^{80} . Pröva nu själv!



```

EDITOR: FAKULTET
PROGRAM LINE 0001
_fakulteten beräknas för n
n = int(input('Skriv in ett positivt heltal:'))
if n >= 0 :
  fakultet = 1
  i = 2
  while i <= n:
    fakultet = fakultet * i
    i=i+1
  print (fakultet)
else:
  print ("Talet måste vara >= 0")

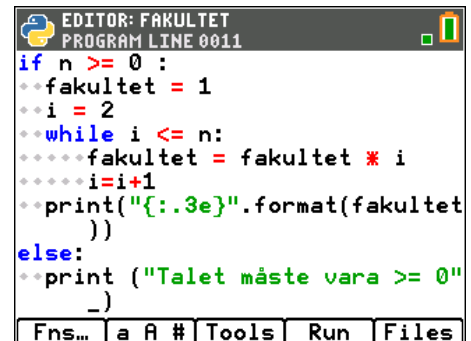
```



```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running FAKULTET
>>> from FAKULTET import *
Skriv in ett positivt heltal:50
30414093201713378043612608166064
76884437764156896051200000000000
0
>>> |

```



```

EDITOR: FAKULTET
PROGRAM LINE 0011
if n >= 0 :
  fakultet = 1
  i = 2
  while i <= n:
    fakultet = fakultet * i
    i=i+1
  print("{:.3e}".format(fakultet))
else:
  print ("Talet måste vara >= 0")

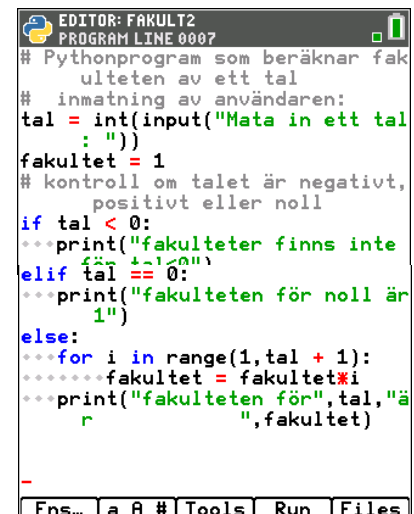
```

Här är ett annat program där vi använder en for-sats istället för while. Programmet är utförligt i den meningen att det "tar hand om" inmatning av talet 0 och har mer förklarande text. Vi har också använt oss av villkorssatser med if-elif-else.

Fundera nu över skillnaden mellan while- och-for-satser för loopar.

Om man i förväg vet hur många gånger man ska upprepa ett block av kod använder man en for-sats.

Testa även detta program för lite större tal och skriv sedan om det så att svaret visas i grundpotensform med 6 decimaler.



```

EDITOR: FAKULT2
PROGRAM LINE 0007
# Pythonprogram som beräknar fakulteten av ett tal
# inmatning av användaren:
tal = int(input("Mata in ett tal : "))
fakultet = 1
# kontroll om talet är negativt, positivt eller noll
if tal < 0:
  print("fakulteter finns inte för +1-0")
elif tal == 0:
  print("fakulteten för noll är 1")
else:
  for i in range(1,tal + 1):
    fakultet = fakultet*i
  print("fakulteten för",tal,"är",fakultet)

```