Dividieren

Wir dividieren mit der Grundschulmethode.

```
n=15                          # n-2 Nachkommastellen
L = [0 for n in range(n)]

Z=2                           # Zähler
N=17                          # Nenner
L[0]=Z

for i in range(n-1):
  r=L[i]%N                    # Rest bei Division, modulo
  L[i]=L[i]//N                # ganzzahlige Division
  L[i+1]=10*r

Zahl=print(str(L[0])+",",end="")
for i in range(1,n-1):
    print(L[i], end="")
```

0,1176470588235

Nun haben wir eine Vorstellung, wie z. B. $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ \dots$
auf viele Stellen ermittelt werden kann.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ \dots$$

Erläutere den Bezug zu $(e^x)' = e^x$.

Setze $x = 1$.

e auf viele Stellen

Wenn wir die Reihe für e nach $1/n!$ abbrechen, so erhalten wir

$$e_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{1}{n!}$$

mit einem Fehler

$$\begin{aligned}
f_n = e - e_n \; &= \; \frac{1}{(n+1)!} + \frac{1}{(n+2)!} + \ldots \\
&= \; \frac{1}{(n+1)!}\Big(1 + \frac{1}{n+2} + \frac{1}{(n+2)(n+3)} + \ldots\Big) \\
&< \; \frac{1}{(n+1)!}\Big(1 + \frac{1}{n+2} + \frac{1}{(n+2)^2} + \ldots\Big) \\
&= \; \frac{\frac{1}{(n+1)!}}{1 - \frac{1}{n+2}} = \frac{1}{(n+1)!}\,\frac{n+2}{n+1} \qquad \text{Summenformel für geom. Reihen}
\end{aligned}$$

Für $n = 70$ gilt $f_n = \dfrac{1}{(n+1)!}\,\dfrac{n+2}{n+1} < 0{,}12 \cdot 10^{-101}$.

Für 100 korrekte Nachkommastellen reicht e_{70},
für 1000 e_{450}, $f_n < 1{,}3 \cdot 10^{-1002}$.

Naheliegende Idee:
Die Fakultät von n wird mit einer Funktion ermittelt.
Die Division $1/n!$ erfolgt in einem Array.
Die Summe wird in einem zweiten Array gebildet.

```
n=70
d=105                          # Arraylänge für 100 Nachkommastellen
L = [0 for n in range(d)]      # für die Summe
M = [0 for n in range(d)]      # für die Division

def f(n):                      # Fakultät
  f=1
  for i in range(2,n+1):
     f=f*i
  return f


L[0]=1                         # der erste Summand der Reihe

for j in range(1,n+1):
    M[0]=1
    m=f(j)

    for i in range(d-1):
      r=M[i]%m                 # Rest bei Division, modulo
      M[i]=M[i]//m             # ganzzahlige Division
      M[i+1]=10*r
      L[i]=L[i]+M[i]

for i in range(d-1,0,-1):      # Wegen der Summation ist L[i]>=10 möglich.
   u=L[i]//10                  # Das wird hier ausgeglichen.
   L[i]=L[i]%10
   L[i-1]=L[i-1]+u


print(str(L[0])+ ",")
for i in range(1,d-4):
      print(L[i], end="")
      if i%5==0:
        print(" ", end=" " )
      if i%25==0:
        print()
```

```
2,
71828  18284  59045  23536  02874
71352  66249  77572  47093  69995
95749  66967  62772  40766  30353
54759  45713  82178  52516  64274
```

e auf viele Stellen

Der vorstehende Algorithmus kann dahingehend verbessert werden, dass die Fakultätsfunktion entfällt und nur noch ein Array erforderlich ist. Rundungsfehler wie bei der Summation werden vermieden.

$$
\begin{aligned}
e_n &= 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} \\
&= 1 + \Big(1 + \Big(1 + \big(1 + (1 + 1 : 4)\big) : 3\Big) : 2\Big) : 1
\end{aligned}
$$

e_n wird ermittelt mit:

```
e=1
while n>0
 e=e/n
 e=e+1
 n=n-1
```

```
n=450
d=1000                      # Nachkommastellen
L = [0 for n in range(d+2)]

while n>0:
    L[0]=1
    for i in range(0,d+1):
      r=L[i]%n              # Rest bei Division, modulo
      L[i]=L[i]//n          # ganzzahlige Division
      L[i+1]=L[i+1]+10*r
    L[0]=L[0]+1
    n=n-1

for i in range(d,0,-1):
  u=L[i]//10
  L[i]=L[i]%10
  L[i-1]=L[i-1]+u


print(str(L[0])+ ",")
for i in range(1,d+1):
      print(L[i], end="")
      if i%5==0:
        print(" ", end="" )
      if i%55==0:
        print()
```

e auf 1000 Nachkommastellen

2,

71828	18284	59045	23536	02874	71352	66249	77572	47093	69995	95749
66967	62772	40766	30353	54759	45713	82178	52516	64274	27466	39193
20030	59921	81741	35966	29043	57290	03342	95260	59563	07381	32328
62794	34907	63233	82988	07531	95251	01901	15738	34187	93070	21540
89149	93488	41675	09244	76146	06680	82264	80016	84774	11853	74234
54424	37107	53907	77449	92069	55170	27618	38606	26133	13845	83000
75204	49338	26560	29760	67371	13200	70932	87091	27443	74704	72306
96977	20931	01416	92836	81902	55151	08657	46377	21112	52389	78442
50569	53696	77078	54499	69967	94686	44549	05987	93163	68892	30098
79312	77361	78215	42499	92295	76351	48220	82698	95193	66803	31825
28869	39849	64651	05820	93923	98294	88793	32036	25094	43117	30123
81970	68416	14039	70198	37679	32068	32823	76464	80429	53118	02328
78250	98194	55815	30175	67173	61332	06981	12509	96181	88159	30416
90351	59888	85193	45807	27386	67385	89422	87922	84998	92086	80582
57492	79610	48419	84443	63463	24496	84875	60233	62482	70419	78623
20900	21609	90235	30436	99418	49146	31409	34317	38143	64054	62531
52096	18369	08887	07016	76839	64243	78140	59271	45635	49061	30310
72085	10383	75051	01157	47704	17189	86106	87396	96552	12671	54688
95703	50354									