

Zadání je napsat programy, kterým se budou předávat parametry a ony budou vracet výsledky, ne provést ad hoc nějaké výpočty.

je potřeba začít programem (procedurou), která vrací počet dní mezi dvěma daty gregoriánského kalendáře. Buď takovou proceduru v matlabu najdete, nebo ji musíte napsat. Pokud ji budete psát, můžete se držet mého algoritmu v maple:

Gregoriánský kalendář

```
> Prestupny := proc (Rok) #procedura zjistí, zda je rok prestupny
Vystup true: je prestupny
```

```
  if Rok mod 4 <> 0
  or Rok mod 100 = 0 and Rok/100 mod 4 <> 0
  then false
  else
  true
  fi
  end ;
```

```
> PocetDniVMesici := proc (Mesic,Rok) #procedura urci pocet dni v
mesici. Vstup je cislo mesice a cislo roku
```

```
  if member( Mesic mod 12, {1,3,5,7,8,10,12,0}) then
  31
  else
  if Mesic = 2 then
  #print(Rok,Prestupny(Rok));
  if Prestupny(Rok) then
  29
  else
  28
  fi
  else 30;
  fi
  fi;
end:
```

```
> DelkaRoku:=proc(Rok)
if Prestupny(Rok) then
366
else
365
fi
end;
```

```
Prestupny := proc (Rok) end proc
  if Rok mod 4 ≠ 0 or Rok mod 100 = 0 and (1/100×Rok) mod 4 ≠ 0
  then false
  else true
  end if
```

```
DelkaRoku :=
```

```
proc (Rok) if Prestupny(Rok) then 366 else 365 end if end proc
```

```
> CisloDne := proc (Den, Mesic, Rokk) #zbytek po deleni sedmi udava den  
v tydnu
```

```
local i;
```

```
i := 'i';
```

```
# korekce pro neexistujici rok 0:
```

```
if Rokk <= 0 then
```

```
Rok := Rokk + 1
```

```
else
```

```
Rok := Rokk
```

```
fi;
```

```
`if` ( Rok >= 2000,
```

```
add(DelkaRoku(i), i = 2000 .. Rok -
```

```
1) + add(PocetDniVMesici(i, Rok), i = 1 .. Mesic - 1) + Den - 2,
```

```
add(-DelkaRoku(i), i = Rok .. 2000 - 1) +
```

```
add(PocetDniVMesici(i, Rok), i = 1 .. Mesic - 1) + Den - 2
```

```
);
```

```
end;
```

```
Warning, `Rok` is implicitly declared local
```

```
CisloDne := proc (Den, Mesic, Rokk)
```

```
local i, Rok;
```

```
  i := 'i';
```

```
  if Rokk <= 0 then Rok := Rokk + 1 else Rok := Rokk end if;
```

```
  `if` (2000 <= Rok, add(DelkaRoku(i), i = 2000 .. Rok - 1)
```

```
    + add(PocetDniVMesici(i, Rok), i = 1 .. Mesic - 1) + Den
```

```
    - 2, add(-DelkaRoku(i), i = Rok .. 1999)
```

```
    + add(PocetDniVMesici(i, Rok), i = 1 .. Mesic - 1) + Den
```

```
    - 2)
```

```
end proc
```

```
>
```

```
> Vzdalenost := (DenA, MesicA, RokA, DenB, MesicB, RokB) ->  
CisloDne(DenB, MesicB, RokB) - CisloDne(DenA, MesicA, RokA);
```

```
> Datum := proc (den)
```

```
local Den, Mesic, Rok;
```

```
Den := den + 2;
```

```
if Den > 0 then
```

```
  i := 0;
```

```
  while Den > DelkaRoku(2000 + i) do
```

```
    Den := Den - DelkaRoku(2000 + i);
```

```
    i := i + 1;
```

```
  od;
```

```
  Rok := 2000 + i;
```

```

i:=1;
while Den > PocetDniVMesici(i,Rok) do
  Den := Den - PocetDniVMesici(i,Rok);
  i:=i+1;
od;
Mesic:=i;
else
#Den:=Den-1;
#print(Den);
i:=0;
while Den <= 0 do
  i:=i-1;
  Den := Den+ DelkaRoku(2000+i);
od;
Rok:=2000+i;
i:=1;
while Den > PocetDniVMesici(i,Rok) do
  Den := Den - PocetDniVMesici(i,Rok);
  i:=i+1;
od;
Mesic:=i;
fi;
# korekce pro neexistujici rok 0:
if Rok<=0 then Rok:=Rok-1 fi;
Den,Mesic,Rok;
end;
Vzdalenost := (DenA, MesicA, RokA, DenB, MesicB, RokB) →
  CisloDne(DenB, MesicB, RokB) – CisloDne(DenA, MesicA, RokA)

```

Warning, `i` is implicitly declared local

```

Datum := proc (den) while DelkaRoku(2000+i) < Den do
local Den, Mesic, Rok, i; Den := Den – DelkaRoku(2000+i); i := i + 1
  Den := den + 2; end do ;
  if 0 < Den then Rok := 2000 + i;
    i := 0; i := 1;
    while PocetDniVMesic(i, Rok) < Den do
      Den := Den – PocetDniVMesic(i, Rok); i := i + 1
    end do ;
    Mesic := i
  else
    i := 0;
    while Den ≤ 0 do
      i := i – 1; Den := Den + DelkaRoku(2000 + i)
    end do ;
    Rok := 2000 + i;

```

```

    i := 1;
    while PocetDniVMesic(i, Rok) < Den do
        Den := Den - PocetDniVMesic(i, Rok); i := i + 1
    end do ;
    Mesic := i
end if ;
if Rok ≤ 0 then Rok := Rok - 1 end if ;
Den, Mesic, Rok
end proc

```

```

>
-730121
31, 12, 1999
-2
31, 12, 1999

```

>

Příklad 1.

Vstup: splákový klendář (asi ve formě souboru, může být třeba tak, že každý řádek je den. měsíc. rok, částka ... to si rozhodněte sami)
a buď velikost dluhu, nebo úroková míra.

Výstup:

úroková míra, nebo velikost dluhu (to co není známo).

Při počítání úrokové míry bude potřeba použít nějakou proceduru pro numerická řešení, numerické hledání kořenů, ...

Úroková míra budiž p. a.

Poznámka: jsou - li z_i splátky v čase t_i a dluh o velikosti Z vznikl v čase t_0 a úroková míra je ξ platí pro okamžik T v němž je dluh splacen, vztah:

> $Z = \sum_{i=1}^T z_i * (1 + \xi)^{((t[0] - t[i]) / \text{DelkaAktualnihoRoku})}, i=1..T);$

$$Z = \sum_{i=1}^T z_i (1 + \xi)^{\left(\frac{t_0 - t_i}{\text{DelkaAktualnihoRoku}} \right)}$$

(Pokud se t počítjí ve dnech).

Příklad 2.

Vstup: splákový klendář (asi ve formě souboru, může být třeba tak, že každý řádek je den. měsíc. rok, částka ... to si rozhodněte sami),
velikost dluhu, úroková míra, čas.

Výstup: zůstatek dluhu v daném čase.

Poznámka: jsou - li z_i splátky v čase t_i a dluh o velikosti Z vznikl v čase t_0 a úroková míra je ξ platí v okamžiku T je zůstatek dluhu:

> $Z \cdot (1 + \xi)^{(T / \text{DelkaAktualnihoRoku})} - \sum_{i=1}^T z[i] \cdot (1 + \xi)^{((T - t[i]) / \text{DelkaAktualnihoRoku})}, i=1..T);$

$$Z(1 + \xi)^{\left(\frac{T}{\text{DelkaAktualnihoRoku}}\right)} - \left(\sum_{i=1}^T z_i (1 + \xi)^{\left(\frac{T - t_i}{\text{DelkaAktualnihoRoku}}\right)} \right)$$

>

Pokud je jednotkou času den.

>

Příklad 3.:

Předpokládáme anuitní měsíční splátky.

Vstup: tři z těchto čtyř veličin:

Velikost dluhu,

velikost měsíčních splátek,

počet let splácení (dvanáctina počtu splátek), nebo počet měsíců plácení,

úroková míra.

Výstup:

chybějící čtvrtá veličina & lineární část přírůstku současné hodnoty splátek (velikosti toho, co si můžeme půjčit) při měně úrokové sazby.

> $r[1] := Z = \text{simplify}(\text{sum}(z \cdot (1 + \xi)^{(-t/12)}, t=1..12 \cdot T));$

$r[2] := Z = \text{simplify}(\text{sum}(z \cdot (1 + \xi)^{(-t/12)}, t=1..N));$

$$r_1 := Z = - \frac{((1 + \xi)^{(-T)} - 1)z}{-1 + (1 + \xi)^{(1/12)}}$$

$$r_2 := Z = - \frac{\left((1 + \xi)^{\left(-\frac{N}{12}\right)} - 1 \right) z}{-1 + (1 + \xi)^{(1/12)}}$$

> $A := \text{subs}(\xi = \xi + \delta, \text{rhs}(r[1]));$ # Castka , kterou si muzeme pujcit, kdyz se urokovka mira zvetsio delta

$B := \text{rhs}(r[1]);$ # castka, kterou si muzeme pujcit pri puvodni urokovke mire

$\text{Prirustek} := \text{simplify}((A - B) / A);$ # relativni prirustk casty, kterou sio budeme moci pujcit, kyz se urokovka mira zvetsi (je-li zvetseni kladne, musi byt aporny)

$\text{LinCPrirustku} := \text{simplify}(\text{coefTayl}(\text{Prirustek}, \delta = 0, 1));$ # jeho linearni cats je koeficient u delta na prvou v taylorove polynomu

$$A := - \frac{((1 + \xi + \delta)^{(-T)} - 1)z}{-1 + (1 + \xi + \delta)^{(1/12)}}$$

$$B := - \frac{((1 + \xi)^{(-T)} - 1)z}{-1 + (1 + \xi)^{(1/12)}}$$

$$\begin{aligned}
\text{Prirustek} := & (1 - (1 + \xi)^{(1/12)} + (1 + \xi + \delta)^T (1 + \xi)^{(1/12)} \\
& - (1 + \xi + \delta)^T (1 + \xi)^{(-T)} + (1 + \xi + \delta)^{(T+1/12)} (1 + \xi)^{(-T)} \\
& - (1 + \xi + \delta)^{(T+1/12)}) / ((-1 + (1 + \xi)^{(1/12)}) (-1 + (1 + \xi + \delta)^T))
\end{aligned}$$

$$\begin{aligned}
\text{LinCPrirustku} := & -(-12 T - 12 (1 + \xi)^{(T+1/12)} T - 2 (1 + \xi)^{(T+1/12)} \\
& + (1 + \xi)^{(2 T+1/12)} + 12 (1 + \xi)^{(1/12)} T + (1 + \xi)^{(1/12)} + 12 (1 + \xi)^T T) \\
& / (12 (1 + \xi) (-1 + (1 + \xi)^{(1/12)}) (-1 + (1 + \xi)^T)^2)
\end{aligned}$$

>

$$-\frac{((1 + \xi)^{(-T)} - 1) z}{-1 + (1 + \xi)^{(1/12)}} = -\frac{((1 + \xi)^{(-T)} - 1) z}{-1 + (1 + \xi)^{(1/12)}}$$

>