

```

[ > #TD 3 Exo 3 :
[ > #Ecrire trois procédures "sommefor", "sommewhile" et "sommerec".
    Ces trois procédures calculent la somme des entiers de 0 à n, n
    étant un entier naturel donné en argument. Ce calcul s'effectue
    à l'aide d'une instruction d'itération for dans le premier cas,
    d'une instruction while dans le second et de façon récursive
    dans le dernier.
[ > sommefor:=proc(n)
    local i,S;
    S:=0;
    for i from 0 to n do
    S:=S+i;
    end do;
    return S;
    end proc;

sommefor :=
    proc(n) local i, S; S := 0; for i from 0 to n do S := S + i end do; return S end proc
[ > sommefor(0),sommefor(1),sommefor(5);
    0, 1, 15
[ > sommewhile:=proc(n)
    local i,S;
    i:=0;
    S:=0;
    while (i<=n) do
    S:=S+i;
    i:=i+1;
    end do;
    return S;
    end proc;

sommewhile :=
    proc(n) local i, S; i := 0; S := 0; while i ≤ n do S := S + i; i := i + 1 end do; return S end proc
[ > sommewhile(0),sommewhile(1),sommewhile(5);
    0, 1, 15
[ > sommerec:=proc(n)
    if n=0 then return 0;
    else return sommerec(n-1)+n;
    end if;
    end proc;

sommerec := proc(n) if n = 0 then return 0 else return sommerec(n - 1) + n end if end proc
[ > sommerec(0),sommerec(1),sommerec(5);
    0, 1, 15
[ > #####
    #Exo 4 : Conversion d'un nombre écrit en base dix dans le
    système de numération des Romains. Pour ce faire il faire il

```

faut coder l'algorithme suivant : soit n l'entier en question et $R(n)$ l'expression représentant l'écriture romaine du nombre n . On écrit l'entier n sous forme de suite de chiffres romains en calculant $R(n)$ par application des règles suivantes :

```
#1000<=n<6000 : R(n)=M, et on passe à R(n-1000);
#900<=n<1000 : R(n)=CM, et on passe à R(n-900);
#500<=n<900 : R(n)=D, et on passe à R(n-500);
#400<=n<500 : R(n)=CD, et on passe à R(n-400);
#100<=n<400 : R(n)=C, et on passe à R(n-100);
#90<=n<100 : R(n)=XC, et on passe à R(n-90);
#50<=n<90 : R(n)=L, et on passe à R(n-50);
#40<=n<50 : R(n)=XL, et on passe à R(n-40);
#10<=n<40 : R(n)=X, et on passe à R(n-10);
#9<=n<10 : R(n)=IX, et on passe à R(n-9);
#5<=n<9 : R(n)=V, et on passe à R(n-5);
#4=n : R(n)=IV, et on passe à R(n-4);
#1<=n<4 : R(n)=I, et on passe à R(n-1);
#0=n : On arrête
```

#Par exemple : $R(1993) \Rightarrow M$, puis $R(993) \Rightarrow CM$, $R(93) \Rightarrow XC$,
 $R(3) \Rightarrow I$, $R(2) \Rightarrow I$, $R(1) \Rightarrow I$, d'où $R(1993) = MCMXCIII$

```
> romain:=proc(n)
local R,r,l,i;
r:=n;
R:=[];
while(r>0) do
if ((1000 <= r) and (r<6000)) then
R:=[op(R),M];
r:=r-1000;
end if;
if ((900<=r) and (r<1000)) then
R:=[op(R),CM]; r:=r-900; end if;
if ((500<=r) and (r<900)) then
R:=[op(R),D]; r:=r-500; end if;
if ((400<=r) and (r<500)) then
R:=[op(R),CD]; r:=r-400; end if;
if ((100<=r) and (r<400)) then
R:=[op(R),C]; r:=r-100; end if;
if ((90<=r) and (r<100)) then R:=[op(R),XC];r:=r-90; end if;
if ((50<=r) and (r<90)) then R:=[op(R),L];r:=r-50;end if;
if ((40<=r) and (r<50)) then R:=[op(R),XL];r:=r-40;end if;
if ((10<=r) and (r<40)) then R:=[op(R),X];r:=r-10; end if;
if (9=r) then R:=[op(R),IX]; r:=r-9;end if;
if ((5<=r) and (r<9)) then R:=[op(R),V]; r:=r-5; end if;
if (4=r) then R:=[op(R),IV]; r:=r-4; end if;
if ((1<=r) and (r<4)) then R:=[op(R),I]; r:=r-1; end if;
```

```

end do;
l:=nops(R);
if l=0 then printf("Il n'y pas d'écriture romaine pour %d\n",n);

else
printf("L'écriture romaine de %d est : ",n);
for i from 1 to l do
printf("%a",R[i]);
end do;
printf("\n");
end if;
end proc;

```

>

```
romain := proc(n)
```

```
local R, r, l, i;
```

```
  r := n;
```

```
  R := [ ];
```

```
  while 0 < r do
```

```
    if 1000 ≤ r and r < 6000 then R := [op(R), M]; r := r - 1000 end if;
```

```
    if 900 ≤ r and r < 1000 then R := [op(R), CM]; r := r - 900 end if;
```

```
    if 500 ≤ r and r < 900 then R := [op(R), D]; r := r - 500 end if;
```

```
    if 400 ≤ r and r < 500 then R := [op(R), CD]; r := r - 400 end if;
```

```
    if 100 ≤ r and r < 400 then R := [op(R), C]; r := r - 100 end if;
```

```
    if 90 ≤ r and r < 100 then R := [op(R), XC]; r := r - 90 end if;
```

```
    if 50 ≤ r and r < 90 then R := [op(R), L]; r := r - 50 end if;
```

```
    if 40 ≤ r and r < 50 then R := [op(R), XL]; r := r - 40 end if;
```

```
    if 10 ≤ r and r < 40 then R := [op(R), X]; r := r - 10 end if;
```

```
    if 9 = r then R := [op(R), IX]; r := r - 9 end if;
```

```
    if 5 ≤ r and r < 9 then R := [op(R), V]; r := r - 5 end if;
```

```
    if 4 = r then R := [op(R), IV]; r := r - 4 end if;
```

```
    if 1 ≤ r and r < 4 then R := [op(R), I]; r := r - 1 end if
```

```
  end do;
```

```
  l := nops(R);
```

```
  if l = 0 then printf("Il n'y pas d'écriture romaine pour %d
```

```
    ", n)
```

```
  else
```

```
    printf("L'écriture romaine de %d est : ", n);
```

```
    for i to l do printf("%a", R[i]) end do;
```

```
    printf("
```

```
      " )
```

```
  end if
```

```
[ end proc
[ > romain(1997);
[ L'écriture romaine de 1997 est : MCMXCVII
[ > romain(2009);
[ L'écriture romaine de 2009 est : MMIX
[ > romain(5789);
[ L'écriture romaine de 5789 est : MMMMDCCLXXXIX
[ >
```