```
> #exercice 01                                                    TP02
> restart;
d := time():  # Démarrer le chronométrage
resultat := sum(i, i = 1..100000);  # Calcul de la somme
temps_ecoule := time() - d;
```

$$resultat := 5000050000$$

$$temps\_ecoule := 0.296$$

```
> restart;
d := time():  # Démarrer le chronométrage
resultat := add(i, i = 1..100000);  # Calcul de la somme
temps_ecoule := time() - d;
```

$$resultat := 5000050000$$

$$temps\_ecoule := 0.031$$

```
> restart;
d := time():  # Démarrer le chronométrage
s:=0;
for i from 1 to 100000 do
s:=s+i;
od:
s;
```

$$s := 0$$

$$5000050000$$

```
> temps_ecoule := time() - d;
```

$$temps\_ecoule := 0.109$$

```
> #exercice 02
> restart;
> E := { a, 2, b, 5, a, c, 7 };
```

$$E := \{2, 5, 7, a, b, c\}$$

```
> whattype(E);
```

$$set$$

```
> E[4];  member( a, E);
```

$$a$$

$$true$$

```
> {}; convert( E, list);
```

$$\{\}$$

$$[2, 5, 7, a, b, c]$$

```
> F := {b, c, d, f, e, h};
```

$$F := \{d, e, f, h, b, c\}$$

```
> E union F;  E intersect F;  E minus F;
```

$$\{2, 5, 7, a, d, e, f, h, b, c\}$$

$$\{b, c\}$$

$$\{2, 5, 7, a\}$$

```
> {2, 7, c} subset E;
```

$$true$$

```
> #exercice 03
> mult := proc(a, b)
    return a * b;
end proc;
```

$$mult := \mathbf{proc}(a, b)\ \mathbf{return}\ a*b\ \mathbf{end\ proc};$$

```
> mult(3, 5);
```

$$15$$

```
> #exercice 04
> #1Somme des entiers de 1 à n :
> sommeEntiers := proc(n)
    local somme, i;
    somme := 0;
    for i from 1 to n do
        somme := somme + i;
    end do;
    return somme;
end proc;
```

$$sommeEntiers := \mathbf{proc}(n)\quad \mathbf{local}\ somme, i;$$
$$somme := 0;\quad \mathbf{for}\ i\ \mathbf{to}\ n\ \mathbf{do}\ somme := somme + i\ \mathbf{end\ do};$$
$$\mathbf{return}\ somme;\quad \mathbf{end\ proc};$$

```
> sommeEntiers(5);
```

$$15$$

```
> #2 Factorielle de n :
> fact := proc(n)
    if n = 0 then
        return 1;
    else
        return n * fact(n - 1);
    end if;
end proc;
```

$$fact := \mathbf{proc}(n)\ \mathbf{if}\ n = 0\ \mathbf{then}\ \mathbf{return}\ 1\ \mathbf{else}\ \mathbf{return}\ n*fact(n - 1)\ \mathbf{end\ if};\ \mathbf{end\ proc};$$

```
> fact(5);
```

$$120$$

```
> #exercice 05

> # Programmer de deux manières la factorielle:
> #1. Factorielle avec récursion:
> factorielle_recursive := proc(n)
    if n = 0 then
        return 1;
    else
        return n * factorielle_recursive(n - 1);
    end if;
end proc;
```

factorielle_recursive := **proc**(*n*)

if *n* = 0 **then return** 1 **else return** *n***factorielle_recursive*(*n* - 1) **end if**;

end proc;

```
> factorielle_recursive(6);
```

720

```
> #Factorielle avec une boucle (itératif):
> fact_iter := proc(n)
    local i, result;
    result := 1;
    for i from 1 to n do
        result := result * i;
    end do;
    return result;
end proc;
```

fact_iter := **proc**(*n*) **local** *i*, *result*; *result* := 1; **for** *i* **to** *n* **do** *result* := *result***i* **end do**; **return** *result*; **end proc**;

```
> fact_iter(6);
```

720

```
> #Exercice 6
> restart;
> Calcul itératif avec une boucle for
> fibonacci:=proc(n)
local u,v,r,k;
u:=1;
v:=1;
for k from 2 to n do r:=u+v; u:=v; v:=r; od;
v;
end;
fibonacci := proc(n)
local u, v, r, k;
u := 1; v := 1; for k from 2 to n do r := u + v; u := v; v := r end do;
end proc
```

fibonacci := **proc**(*n*)
local *u*, *v*, *r*, *k*;
u := 1;
v := 1;
for *k* **from** 2 **to** *n* **do** *r* := *u* + *v*; *u* := *v*; *v* := *r*; **end do**;
v;
end proc;

```
end proc;
```

```
fibonacci := proc(n)
local u, v, r, k;
      u := 1;
      v := 1;
      for k from 2 to n do r := u + v; u := v; v := r; end do;
end proc;
```

```
> fibonacci(5);
```

8

```
> #Calcul récursif
```

```
> restart;
```

```
> fiborecursif:=proc(n)
  local r;
  if n=0 then r:=1; elif n=1 then r:=1; else
  r:=fiborecursif(n-1)+fiborecursif(n-2); fi;
  r;
  end;
```

```
fiborecursif := proc(n)
local r;
      if n = 0 then r := 1 elif n = 1 then r := 1 else r := fiborecursif(n - 1) + fiborecursif(n - 2) end
      r;
end proc;
```

```
> fiborecursif(5);
```

8

```
> t1:=time():fibonacci(30);time()-t1;
```

1346269

0.

```
> t2:=time():fiborecursif(30);time()-t2;
```

1346269

8.080

```
>
```