Tri Selection:

```
procedure Selection(n: entier, @ t : tab):
    debut
        pour i de 1 à n-1 faire
            j <- i
            tmp <- t[i]
            tant que j > 1 et tmp < t[j-1] faire
                t[j] <- t[j-1]
                j <- j-1
            fin tant que
            t[j] <- tmp
        Fin pour
    Fin
```

Var	type
i	entier
j	entier
tmp	entier ou réel !

Tri à bulle:

```
procedure bubble(n: entier, @ t : tab):
    Debut
        pour i de 0 à n-2 faire:
            Si t[i] > t[i+1] alors
                tmp <- t[i]
                t[i] <- t[i+1]
                t[i+1] <- tmp
                bubble(n, t)
            Fin Si
        Fin pour
    Fin
```

Var	type
i	entier
tmp	entier ou réel.
bubble	procedure

Tri Shelle:

```
Procedure Shell (n: entier, @ t: tab):
   Debut :
      Pas ← 1
      tant que pas × 3 + 1 ≤ n faire
            pos ← pos × 3 + 1
      Fin tant que.
      Repeter:
      Pour i de pas a n-1 faire
            tmp ← t[i]

            j ← i

            tant que j ≥ 1 et t[j - pos] < tmp faire
                  t[j] ← t[j - pos]

                  j ← j - pas

            Fin tant que
            t[j] ← tmp

      Fin pour

      Pas ← pos div 3
      Jusqua' a (pas ≤ 1)

   Fin
```

project". Mathcomplex:.

Cartisienne:.

Def func colci $(2-2i)^2$

$$a^2 + 2ab + b^2$$

function colc (reg:String):String:

Begin

// separate",

13/12/24:

Correction devoir Synth n°1:

P.P

Algorithme point-equilibre∫
   Début
      Saisir(l,c)
      fillmat(l,c,m)
      src ← "equilibre.dat"
      remplir∫(m,src,l,c)
      src₁ ← "mon_Somme.txt"
      remplir∫t(src,src₁,m)
   Fin

TDO :

Nom	type
l,c	entier
src, src₁	chaîne
m	mot

Sonction(m: mot, l: entier, c: entier): boolean;
   Début
      i ← -1
      Repeter
         i ← i+1
         chek ← $m[l,i] \neq m[i,c]$
      Jusqu'à chek = faux ou i = c-1
      Retourner chek.
   Fin

TDO :
i: entier
chek: bool

Recursivités:

Fonction fact (n: entier) : entier

Début
    Si n=0 alors f←1

Sinon

      f←1
      pour i de un a (pos-1) faire
          f← f x i

Fin Si pour
retourner f

Fin

Fonction rec (n: entier) : entier

Début
    Si n=0 alors
      retourner 1

Sinon
    f←1

Procedure remplir Pb (@

S_1

$S \geq S_1$ où S_2

col := σ

Fonction (m : mat, P : entier, Pi : entier, Pf : entier) : entier
   Debut
      $S \leftarrow 0$
      Pour i de Pi a Pf faire
         $S \leftarrow S + m[P, i]$
      Fin pour
      retourner S
   Fin

TDO
───────────
i : entier

S : entier