Algo :

- bac 2019 :

Exo 1 :

```
Fonction what (a, b : réel) : réel
Debut
  si (a - b) >= 0 alors
    retourner a
  Sinon
    retourner what (b, a)
  Fin Si
Fin
```

① → Réel

② → (a - b) ≥ 0

③ → 12

④ → C

Exo 2 :

$f(x) = \frac{1}{x}$, $d \geq]0; +\infty[$: / $\int_1^a f(x)\,dx$:

1) Fonction surface (a : réel, n : entier) : réel // Rectangle a gauche :

```
Debut
  S ← 0
  x ← 1
  h ← (a - 1)/n
  pour i de 1 a n faire
    S ← S + 1/x
    x ← x + h
  fin pour
  retourner S x h
Fin
```

TDO		
S	réel	surface $\int_1^a f(x)\,dx$
x	réel	Val d'abscisse
h	réel	Longeur
i	entier	compteur

2) fonction Calcul (a : réel, n : entier) : réel.

```
Debut
  eps ← 0,0001
  e ← a
  tantque abs(surface(a,n) - 1) > eps faire
      e ← e + eps
  Fin tantque
  retourner e
Fin
```

to check !

Var	type
e	réel
eps	réel
surface	fonction

Exo 3

```
Fonction pgcd(a, b : entier) : entier
  Debut
    r ← a mod b
    si r = 0 alors
      retourner b
    fin si
    retourner pgcd(b, r)
  Fin
```

Var	type
r	entier

```
procedure Inv():
  Debut
    ouvrir(source, "Fraction.dat", "rb")
    ouvrir(Res, "Irreduct.dat", "wb")
    tantque non fin_fichier(source)
      lire(source, objet)
      Si pgcd(objet.Num, objet.Denom) = 1 alors
        ecrire(Res, objet)
      Fin si
```

TDOL

Var	type
source	fich
Res	fich
objet	frac
pgcd	fonction

fermer (Res)

fermer (source)

Fin

TDONT

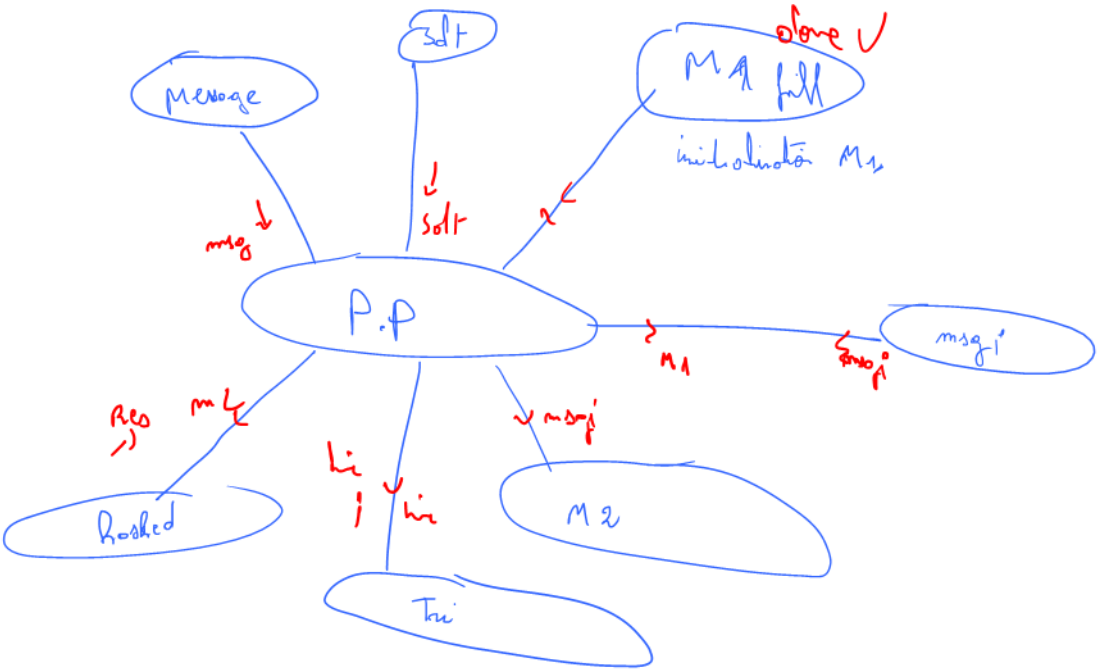
frac = enregistrement

Num : entier

Denom : entier

fich = fichier de type frac

problème



Algorithme Hash

Debut

Message ← msg( )

salt ← key( )

initialisation(M1)

Message2 ← msgi(M1)

FillM(M2, Message2, salt)

Sort(M2[0])

Res ← hashed(M2)

Fin

TDOG	
objet	type
Message, message2, salt, Res	chaine de charactere
M1	mot 1
M2	mot 2
msg, key, msgi, hashed	Fonction
initialisation, FillM, sort,	procedure

TDONT G

M1 : tableau 6x6 de charactere

M2 : tableau 6x7 de charactère

---

module Saisie de message :

Fonction msg( ) : chaine

Debut

Repeter

good ← Vrai

Lire(ms)

pour i de 0 à Long(ms) - 1 faire

si ord(ms[i]) ∉ [97, 122] alors

good ← faux

fin si

fin pour

good ← Long[ms] ≤ 18

jusqu'a good

Retourner ms

Fin

TDOL	
good	booleen
ms	chaine

Fonction soft( ) : chaine de charactère :

Debut

good ← vrai

Repeter

lire (s)

pour i de 0 à Long(s) - 1 faire

si ord [s[i]) ∉ [ord("A"); ord("B")] alors

good ← faux

fin si

Fin pour

good ← Long(s) = 6

jusqu'à good

retourner s

Fin

TDOL	
objet	type
s	chaine de chara
good	booleen

Fonction msg1 (M : mot1) : chaine de charactère

Debut

0 1 2 3 4 5

A B C D E F

1 2 3 4 5 6

ord A = 65

ord Z = 90