

Alopa !

• bac 2019 :

ex 1 :

Fonction $\text{whot}(a, b : \text{réel}) : \text{réel}$

Debut

si $(a - b) \geq 0$ alors

retourner a

sinon

retourner $\text{whot}(b, a)$

Fin si

Fin

① \rightarrow réel

② $\rightarrow (a - b) \geq 0$

③ $\rightarrow 12$

④ $\rightarrow c$

ex 2 :

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

1) fonction surface $(a : \text{réel}, m : \text{entier}) : \text{réel}$ // Rectangle à gauche :

Debut

$s \leftarrow 0$

$xc \leftarrow 1$

$h \leftarrow (a - 1) / m$

Pour i de 1 à m faire

$s \leftarrow s + 1 / xc$

$xc \leftarrow xc + h$

fin pour

retourner $s \times h$

Fin

TDD		
s	réel	surface $\int_1^a f(x) dx$
xc	réel	val d'abscisse
h	réel	Longeur
i	entier	compteur

2) fonction Cdeal (a: réel, m: entier): réel.

Début

$\epsilon \leftarrow 0,0001$

$e \leftarrow a$

tant que $\text{abs}(\text{surface}(a, m) - 1) > \epsilon$ faire

$e \leftarrow e + \epsilon$

Fin tant que

retourner e

Fin

to check!

Var	type
e	réel
eps	réel
surface	fonction

esc 3

Fonction pgcd (a, b: entier): entier

Début

$n \leftarrow a \bmod b$

si $n = 0$ alors

retourner b

fin si

retourner pgcd (b, n)

Fin

Var	type
n	entier

procédure Inn():

Début

ouvrir (source, "Fonction.dot", "wb")

ouvrir (Res, "Ineduct.dot", "wb")

tant que non fin.fichiers (source)

lis (source, objet)

$p \leftarrow \text{pgcd}(\text{objet}.\text{num}, \text{objet}.\text{denom})$

$p \leftarrow \text{objet}.\text{num} / p$ $p \leftarrow \text{objet}.\text{denom} / p$

tant que $p \neq 1$ faire

TDOL

Var	type
source	file
Res	file
objet	file
pgcd	fonction

0 'er

$f \subseteq \text{to home}$
 $\text{Envie}(\text{Res},$

$\text{fermer}(\text{Res})$

$\text{fermer}(\text{source})$

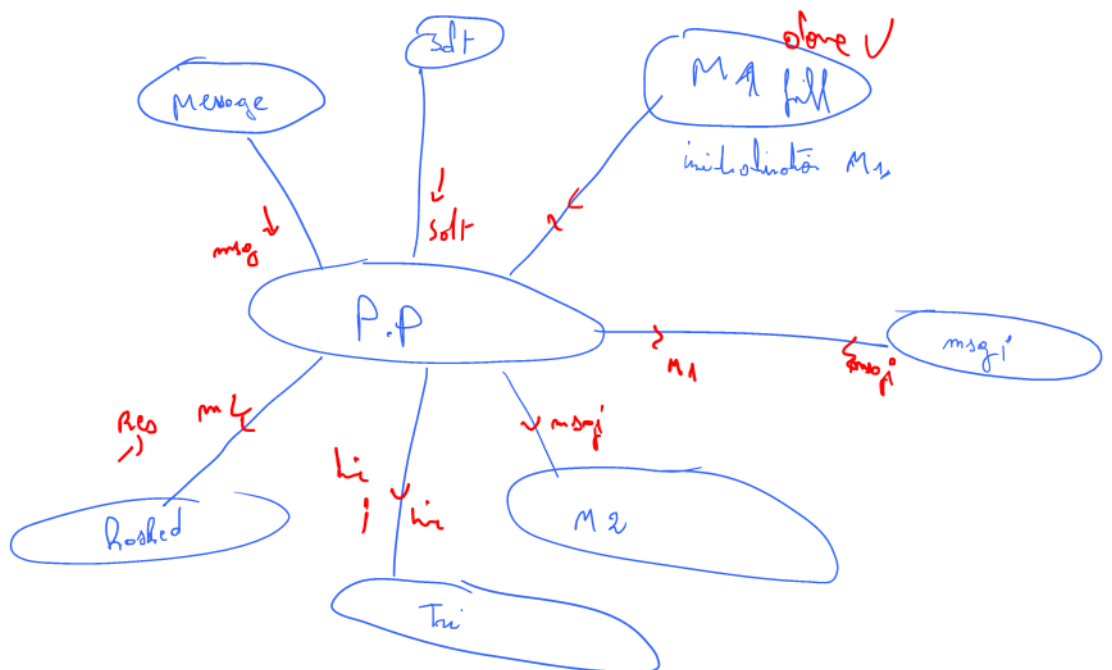
Fin

probleme

TD 0 NT

$\text{frac} = \text{enregistrement}$
 $\text{Num} : \text{entier}$
 $\text{Denom} : \text{entier}$

$\text{fich} = \text{fichiers de type frac}$



Algorithme Hash

Debut

Message \leftarrow msg()

salt \leftarrow key()

initialisation(M_1)

Message2 \leftarrow msgi(M_1 , Message)

FillM(M_2 , Message2, salt)

sort(M_2)

Res \leftarrow hashed(M_2)

Fin

TDOG	
objet	type
Message, Message2, Salt, Res M_1 M_2	chaîne de caractères mot 1 mot 2
msg, key, msgi, hashed initialisation, FillM, sort,	Fonction procédure

TDO NT G
M_1 : tableau 6x6 de caractères M_2 : tableau 6x7 de caractères

module Saisir de messages

Fonction msg() : chaîne

Debut

Repete

good \leftarrow vrai

lire(ms)

pour i de 0 à Long(ms) - 1 faire

si ord(ms[i]) \notin [97, 122] alors

good \leftarrow faux

fin si

fin pour
good \leftarrow Long[ms] \leq 18

jusqu'à good

Retourner ms

Fin

TDOL	
good ms	boolean chaîne

Fonction $\text{sort}()$: chaîne de caractère :

Début

$\text{good} \leftarrow \text{vrai}$

Repete

$\text{vie}(s)$

Pour i de 0 à $\text{long}(s) - 1$ faire

si $\text{ord}(s[i]) \notin [\text{ord}('A'); \text{ord}('Z')]$ alors

$\text{good} \leftarrow \text{faux}$

fin si

Fin pour

$\text{good} \leftarrow \text{long}(s) = 6$

jusqu'à good

retourner s

Fin

TDOL	
objet	type
s	chaîne de caractères
good	booléen

Fonction $\text{msg}_i (m : \text{mot})$: chaîne de caractère

Début

0 1 2 3 4 5

A B C D E F

1 2 3 4 5 6

$\text{ord} A = 65$
 $\text{ord} Z = 90$

$$65 + 0 = A$$

$$65 + 1 = B$$

$$65 + 2 = C$$

$$65 + 3 = D$$

$$65 + 4 = E$$

$$65 + 5 = F$$

Fonction msgi(m, mot1, msg) : chaîne

Debut

Res ← ""

pour i de 0 à Long(msg) - 1 faire

a ← 0

b ← 0

quit ← Faux

Si msg[i] ≠ " " alors

tant que (a < 6 et b < 6) ou non quit faire

Si msg[i] = m[a, b] alors

Res ← Res + chr(ord("A") + b) + chr(ord("A") + a)

quit ← Vrai

Fin Si

b ← b + 1

Si b = 6 alors

a ← a + 1

b ← 0

Fin tant que

Si non

Res ← Res + msg[i]

Fin Si

Fin pour

retourner Res

Fin

TNO L	
objet	type
a, b, i	entier
Res	chaîne

Procédure FillM(m : mot 2, msg : chaîne, $solt$: chaîne)

Début

pour k de 0 à 5 faire

$m[0, k] \leftarrow solt[k]$

fin pour

$k \leftarrow 0$

$i \leftarrow 1$

$f \leftarrow 0$

tant que $k < \text{Long}(msg)$ ou ($i < 7$ et $f < 6$) faire

$m[i, f] \leftarrow msg[k]$

$k \leftarrow k + 1$

$f \leftarrow f + 1$

Si $f = 6$ alors

$i \leftarrow i + 1$

$f \leftarrow 0$

Fin Si

Fin tant que

Fin

TPOOL	
obj	type
i, f, k	entier

procedure sort (m: mot 2):

Debut

// bubble!

isSorted ← false

tant que non isSorted faire

isSorted ← vrai

— pour i de 0 à 4 faire

si ord(m[0,i]) > ord(m[0,i+1]) alors

swapp ← m[0,i]

m[0,i] ← m[0,i+1]

m[0,i+1] ← swapp

isSorted ← false

Fin si

Fin pour

Fin tant que

Fin

obj	type
isSorted	booléen
i	entier
swapp	chaîne.

Fonction hashed (m: mot 2): chaîne

Debut

Res ← ""

pour i de 0 à 6 faire

pour j de 0 à 5 faire

Res ← Res + m[i,j]

Fin pour

Fin pour

retourner Res

Fin

TMDL	
obj	type
i, j	entier
Res	chaîne

