

Aléa (^{entier} début , ^{entier} Fin)

Aléa ("A" , "Z")

^{Caractère} C ← Chr (Aléa (ord("A"), ord("Z")))
_{entier · code ASCII}

C ← passage par valeur

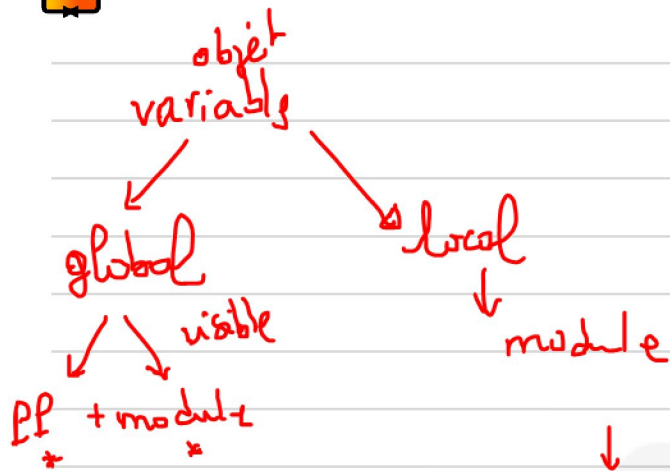
C ← "Bonj" C ← "or" ⇒ C = "Bonjour"

Procédure Permuter (^{a: entier, b: entier} a, b: entier)
 début _{(b, a: entier) & !!}

^a 65 ^b 18⁵
 (65) (18)

{ x ← a
 a ← b
 fin b ← x
 }
 Def Permuter () :
 global a, b
 ≡ x = a
 a = b
 b = x
 return a, b

a → return
 a → global



objet	visible PF	visible module
x		X
A	X	X
c		X

• $ch \leftarrow \text{"Infinisique"}$

Annotations: 1 points to 'In', 2 points to 'finisique'.

$ch1 \leftarrow \text{sous chaîne}(ch, 2, 8) \Rightarrow ch1 = ch[2:8]$

• $ch2 \leftarrow \text{effacer}(ch, 2, 8)$

$ch1 \leftarrow ch[0:2] + ch[8:]$

$ch[:2]$

Code	1234	1235	1237
Etat	"L"	"U"	"L"

C = "1235"

* procédure Saisir (a) n: entier

début

Répéter

lire (n)

jusqu'à $10 \leq n \leq 100$

fin

* procédure Remplir (a) Code: tab ; n: entier

début

pour i de 0 à n-1 faire

Répéter

lire (code [i])

ch ← convch (code [i])

jusqu'à $(\text{long}(\text{ch}) = 13)$ et $(\text{pair}(\text{sous-chaine}(\text{ch}, 0, 3)))$

et $(\text{suite}(\text{ch}))$

fin pour

fin

T.D.O.L

i	entier
ch	chaîne
pair	fonction.
suite	

fonction pair (ch: chaîne) : booléen
début

$x \leftarrow \text{valeur}(ch)$

$i \leftarrow 2$

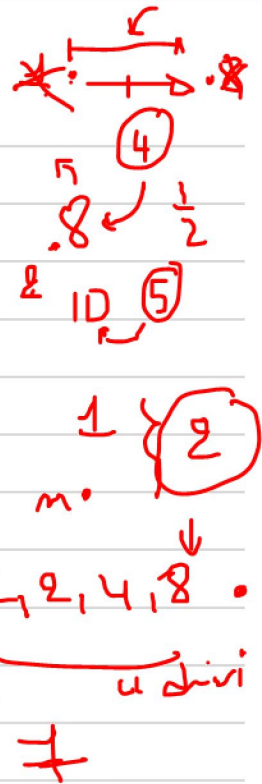
Tant que ($x \bmod i \neq 0$ et $i \leq n \text{ div } 2$)

$i \leftarrow i + 1$

fin Tant que

retourner $x \bmod i \neq 0$

Fin



fonction pair (ch: chaîne) : booléen
début

$x \leftarrow \text{valeur}(ch)$

$n \leftarrow 0$

pour i de 1 à n faire

si ($x \bmod i = 0$) alors

$n \leftarrow n + 1$

fin si

fin pour

retourner $n = 2$

fin



fonction suite (ch: chaîne): boolean
début

OK ← fause

$i \leftarrow 0$

Tant que ($i < \text{long}(ch) - 4$ et OK = fause)

a = valeur (ch[i])

b = valeur (ch[i+1])

c = valeur (ch[i+2])

d = valeur (ch[i+3])

si (b-a = c-b et (c-d = d-c)) Alors
OK = vrai

sinon $i \leftarrow i + 1$

fin Tant que

retourner OK

Fin

procédure Initialiser (a: état; tab: tableau; n: entier)

début

pour i de 0 à n-1 faire

Etat[i] ← "L"

fin pour

fin

procédure lecture (a) c : entier)

début

Répéter

lire (c)

cha ~~converch~~(c)

jusqu'à ($\text{long}(c) = 13$) et pair (sous chaîne (ch, 0, 3)
et suite (ch)

fin

fonction Recherche (c : entier, Code : tab; m : entier) : ^{entier}

début

ok ← faux

i ← 0 ; $p \leftarrow -1$

Répéter

si ($c = \text{code}[i]$) alors

ok = vrai ; $p \leftarrow i$

sinon

i ← i + 1

Fini

jusqu'à ($\text{ok} = \text{vrai}$) ou ($i = n - 1$)

retourner ~~ok~~ p

fin

procédure Affichage (c: entier; code: tab; m: entier; Etat: tab)
 début
 $p \leftarrow Recherche(c, code, m)$

Si (Recherche (c, code, m) = -1) Alors
 | écrire ("le code est inexistant")

Sinon si (Etat[p] = "V")
 | écrire ("code utilisé")

Sinon

| écrire ("opération de recharge réussie")
 | Etat[p] ← "V"

Fin Si

Fin

P P

début

saisir (m)

Remplir (code, m)

Initialiser (Etat, m)

lecture (c)

Affichage (c, code, m, Etat)

Fin