

TD3 : les procédures et les fonctions

Ex 1 :

Procédure SaisieBornes (var A : entier,
var B : entier)

variable

debut

Repéter

écrire ("donner la borne inf")

lire (A)

Jusqu'à (A > 0)

Repéter

écrire ("donner la borne sup")

lire (B)

Jusqu'à (B > A)

Fin

Procédure CalcProd (A : entier,
B : entier, var P : entier)

variable

i : entier

Debut

P ← 1

pour i de A à B faire

P ← P * i

Fin pour

Fin

Procédure Affichage (A : entier, B : entier
P : entier)

variable

Debut

Ecrire (" .. Bi = ", A)

Ecrire (" .. B_s = ", B)

Ecrire (" .. P = ", P)

Ecrire (" .. A, et B est ", P)

Fin

a)

Algorithme ProduitIntervalle

Var

B_i, B_s, P : entier

debut

Saisie Borne (B_i, B_s)

CalculeProduit (B_i, B_s, P)

Affichage (B_i, B_s, P)

Fin

b)

Fonction CalcProd (A : entier, B : entier) : entier

Var

P, i : entier

debut

P ← 1

pour i de A à B faire

P ← P * i

Fin Pour

retourner P // ou bien CalcProd ← P

Fin

b) Algorithme ProduitIntervalle3

Var

B_i, B_s, P : entier

debut

Saisie-Borne (B_i, B_s)

P ← CalcProd (B_i, B_s)

Affichage (B_i, B_s, P)

Fin

Ex 3 :

Fonction Min (A : réel, B : réel) : réel

Var

debut

si (A < B) alors Min ← A

sinon Min ← B

fin si

Fin

Fonction Max (A: réel, B: réel): réel

Var

Debut

si ($A > B$) alors retourner (A)

sinon retourner (B)

Fini

Fin

Algorithme ex 3

Var

x, y, z, w, M₁, M₂, M : réel

Debut

écrire ("donner 4 nombres")

lire (x, y, z, w)

M₁ \leftarrow Min (x, y)

M₂ \leftarrow Min (z, y)

M \leftarrow Min (M₁, M₂)

écrire ("Min =", M)

écrire ("Max =", Max(Max(x, y),
Max(z, w)))

Fin

Exercice 4 :

i) Procédure saisie (var n: entier)

Var

Debut

Saisie (n)

Repter

Ecrire ("donner un entier")

Lire (n)

Jusqu'à ($n > 0$)

Fin

Fonction somme (n: entier): entier

Var

s: entier

Debut

s \leftarrow 0

Repter

s \leftarrow s + n mod 10

n \leftarrow n div 10

Jusqu'à ($n = 0$)

retourner (s)

Fin

Fonction Product (n: entier): entier

Var

P: entier

Debut

P \leftarrow 1

Repter

si ($n \text{ mod } 10 \neq 0$) alors

p \leftarrow p * n mod 10

Fini

n \leftarrow n div 10

Jusqu'à ($n = 0$)

retourner (P)

Fin

Algorithme ex 4

Var

A, B: entier

Debut

Saisie (A)

Saisie (B)

si (Somme (A) = Somme (B)) alors

écrire (A, "et", B, "sont amis")

sinon

écrire (A, "et", B, "ne sont pas amis")

Fini

si (produit (A) = Produit (B)) alors

écrire (A, "et", B, "sont cousins")

sinon

écrire (A, "et", B, "ne sont pas cousins")

Fin

Fini

Ex 6:

Fonction saisie () : entier

Var

\bullet x : entier

Debut

Repete

ecrire ("donner un entier")

Lire (x)

Jusqu'à ($x > 9$ et $x < 100$ et
 $x \bmod 10 \neq x \cdot \text{div} 10$)

saisie $\leftarrow x$

Fin

Procedure liste_g (x : entier)

Var

I, dif : entier

Debut

ecrire (x)

Repete

I $\leftarrow (x \bmod 10) * 10 + x \cdot \text{div} 10$

dif $\leftarrow \text{Abs}(x - I)$

ecrire (dif)

x \leftarrow dif

Jusqu'à (dif = 9)

Fin

Algorithme ex 6

Var

A : entier

Debut

A \leftarrow saisie ()

liste_g (A)

Fin

Ex 7:

Fonction somme (i : entier) : reel

Variable

s : reel j : entier

Debut

S $\leftarrow 0$

Pour j de 1 à i faire

S $\leftarrow S + \frac{1}{j}$

Fin Pour
retourner (S)

Fin

Fonction Produit (n : entier) : reel

Variable

P : Reel

i : entier

Debut

P $\leftarrow 1$

Pour i de 1 à n faire

P $\leftarrow P * \text{somme}(i)$

Fin Pour

retourner (P)

Fin

Algorithme ex 7

Var
n : entier

Debut

Repete

ecrire (" ")

Lire (n)

Jusqu'à (n > 0)

ecrire ("Produit = ", Produit(n))

Fin

Exercice 8:

1) procédure racine (var x: réel)
variable

Debut

Repéter
écrire (" ")
lire (x)
Jusqu'à ($\alpha > 0$)

Fin

Fonction Racine (x: réel)

variable

Debut d, U0, U1 : réel

$U_0 \leftarrow (1 + x) / 2$

Repéter

$U_1 \leftarrow (U_0 + x/U_0) / 2$

$d \leftarrow U_1 - U_0$

$U_0 \leftarrow U_1$

Jusqu'à ($|d| \leq 10^{-4}$)

retourner (U_1)

Fin

Exercice 9:

Algorithme exc8

variable

R, x : réel

Debut

Saisie (x)

$R \leftarrow \text{Racine}(x)$

écrire (R)

Fin

Exercice 9:

Ex 9:

Fonction Calcul (n: entier): entier
variable

i, U0, U1, U : entier

Debut

si ($n = 0$) alors retourner (0)

sinon

si ($n = 1$) alors retourner (1)

sinon

retourner (calcul(n-1) + calcul

(n-2))

Fin

Fin si

Fin

Fin si

$U_0 \leftarrow 0, U_1 \leftarrow 1$

si $n = 0$ alors retourner (U_0)

sinon

si ($n = 1$) retourner (U_1)

sinon

pour i de 2 à n faire

$U \leftarrow U_1 + U_0$

$U_0 \leftarrow U_1$

$U_1 \leftarrow U$

Fin Pour

retourner (U)

Fin