



Correction de la série de TD 3

Exercice 1

Algorithme Diviseurs ;
Variable n, i : Entier ;

Début

Ecrire ("Entrer un entier positif=") ; Lire(n) ;

Pour i ← 1 à n **Faire**

Si (n mod i == 0) **Alors** /* Si le reste de la division de n par i est égale à 0 */
 Ecrire(i) ;

Fin Si

FinPour

Fin

Exercice 2

• Version Pour...Faire

Algorithme Facto ;

Variable

n, i, f : Entier ;

Début

Ecrire ("Entrer un entier positif=") ; Lire(n) ;

f ← 1 ; /* initialisation de la factorielle à 1 puisque 1! = 1 */

Pour i ← 2 à n **Faire**

 f ← f * i ; /* Pour chaque parcours on multiplie l'ancienne valeur de f par i */

FinPour

Ecrire (n, "! =", f)

Fin.

• Version Tantque ...Faire

Algorithme Facto ;

Variable

n, i, f: Entier ;

Début

Ecrire ("Entrer un entier positif =") ; Lire(n) ;

f ← 1 ; /* initialisation de la factorielle à 1 puisque 1! = 1 */

i ← 2 ; /* initialisation du compteur i */

Tantque (i ≤ n) **Faire**

 f ← f * i ; /* Pour chaque parcours on multiplie l'ancienne valeur de f par i */

 i ← i + 1 ; /* incrémentation du compteur i */

FinPour

Ecrire (n, "! =", f) ;

Fin.

• **Version Répéter...TantQue**

Algorithme Facto ;

Variable

n, i, f : Entier ;

Début

Ecrire ("Entrer un entier positif="), Lire(n) ;

f ← 1 ; /* initialisation de la factorielle à 1 puisque 1! = 1 */

i ← 2 ; /* initialisation du compteur i */

Répéter

f ← f * i ; /* Pour chaque parcours on multiplie l'ancienne valeur de f par i */

i ← i + 1 ; /* incrémentation du compteur i */

TantQue (i ≤ n)

Ecrire (n, "! =", f) ;

Fin.

Exercice3

Algorithme Notes ;

Variable n, i: Entier ;

note, nmin, nmax, s : Réel ;

Début

Ecrire ("Entrer le nombre de notes=") ;

Lire(n) ; /* *On suppose que n est toujours supérieur à zéro* */

s ← 0 ; nmin ← -20 ; nmax ← 0 ;

Pour i ← 1 à n **Faire**

Ecrire ("Entrer une note=") ; Lire (note) ;

s ← s + note /* *additionner la nouvelle note* */

Si (note < nmin) **Alors**

nmin ← note ; /* *mémorisation de la nouvelle valeur minimale* */

Fin Si

Si (note > nmax) **Alors**

nmax ← note ; /* *mémorisation de la nouvelle valeur maximale* */

Fin Si

FinPour

Ecrire ("Meilleure note=", nmax) ;

Ecrire ("Mauvaise note=", nmin) ;

Ecrire ("Moyenne des notes =", s/n) ;

Fin

Exercice 4

Algorithme Puissance ;

Variable a, c: Réel ;

b, i: Entier ;

Début

Ecrire ("Entrer la valeur de a=") ; Lire(a) ;

Ecrire ("Entrer la valeur de b=") ; Lire(b) ;

c ← 1 ; /* *initialisation du résultat du produit* */

Pour i ← 1 à Abs(b) **Faire**

c ← c * a ; /* *produit de axa... b fois* */

FinPour

Si ($b < 0$) **Alors** (*si b est négative alors le résultat sera 1/c*)

$c \leftarrow 1 / c$;

Fin Si

Ecrire (a, "à la puissance", b, "=", c) ;

Fin

Exercice 5

Algorithme Premier ;

Variable n, i, nb_div: Entier ;

Début

Ecrire ("Entrer un entier positif=") ; Lire(n) ;

$nb_div \leftarrow 0$; /* initialisation du nombre de diviseurs */

$i \leftarrow 2$;

Tantque($i \leq n \text{ div } 2$) **Faire**

Si ($n \text{ mod } i == 0$) **Alors**

$nb_div \leftarrow nb_div + 1$; /*incrémentation du nombre de diviseurs */

FinSi

$i \leftarrow i + 1$;

Fin Tantque

Si ($nb_div == 0$) **Alors**

Ecrire (n, " est un nombre premier") ;

Sinon

Ecrire (n, " n'est pas un nombre premier") ;

FinSi

Fin

Exercice 6

Algorithme PGCD ;

Variable a, b: Entier ;

Début

Ecrire ("Entrer la valeur de a=") ; Lire(a) ;

Ecrire ("Entrer la valeur de b=") ; Lire(b) ;

Répéter

Si ($a > b$) **Alors** $a \leftarrow a - b$;

FinSi

Si ($a < b$) **Alors** $b \leftarrow b - a$;

FinSi

Tantque ($a \neq b$)

Ecrire ("Le PGCD=", a) ;

Fin

Exercice 7

Algorithme PPCM ;

Variable a, b, t, p: Entier ;

Début

Ecrire ("Entrer la valeur de a=") ; Lire(a) ;

Ecrire ("Entrer la valeur de b =") ; Lire(b) ;

Si (a<b) **Alors**

 t←a ; /*-----*/

 a←b ; /* Permutation */

 b ←t ; /*-----*/

FinSi

p ←a ;

Tantque ((p mod b) !=0) **Faire**

 p ←p+a ;

Fin Tantque

Ecrire ("PPCM =", p) ;

Fin

Exercice 8

Algorithme Fibon ;

Variable

 F0, F1, F, i : Entier;

Début

F0 ←1;

Ecrire ("F0 =", F0) ;

F1 ←1 ;

Ecrire ("F1 =", F1) ;

Pour i ←2 à 9 **Faire**

 F←F0+F1 ;

 Ecrire ("F", i, " =",F);

 F0 ←F1;

 F1 ←F;

FinPour

Fin.

Exercice 9

Algorithme Somme ;

Variable N , i : Entier ;

 s : Réel ;

Début

Ecrire ("Entrer la valeur de n=") ; Lire(n) ;

s←0 ; /* initialisation de la somme */

Pour i ←1 à n **Faire**

 s ← s +1.0/i ; /* ajouter à l'ancienne somme la nouvelle valeur de1.0/i */

FinPour

Ecrire ("somme=", s) ;

Fin

Exercice 10

Algorithme cubique ;

Variable i, centaine, dizaine, unite : Entier ;

Début

Pour i ← 150 à 410 **Faire**

 centaine ← i div 100 /* ex : i=150 + centaine=1 */

 dizaine ← (i mod 100) div 10 : /* ex : i=150 + dizaine=5 */

 unite ← (i mod 100) mod 10 ; /* ex : i=150 + unite=0 */

Si ((centaine³ + dizaine³ + unite³) ==i) **Alors**

 Ecrire (i, " est un nombre cubique") ;

Fin Si

FinPour

Fin

Remarque: les nombres cubiques à trouver sont : 153, 370, 371 et 407

Exercice 11

Algorithme parfaits ;

Variable i, n, s, j: Entier ;

Début

Pour i ← 2 à 1000 **Faire**

s ← 0 ;

Pour j ← 1 à (i div 2) **Faire**

Si (i mod j == 0) **Alors**

 s ← s + j;

Fin Si

FinPourj

Si (s=i) **Alors** Ecrire (i, " est un nombre parfait") ;

Fin Si

FinPouri

Fin