

Exercice 1.

Écrire un algorithme qui permet de lire deux nombres, calculer la somme et le produit, et afficher si ces derniers sont positifs ou négatifs.

Exercice 2.

Donner un algorithme qui permet de lire un nom d'utilisateur (une chaîne) et vérifie s'il s'agit de la chaîne "hamid". Il affiche ensuite "Bienvenue Hamid". Il lui demande de saisir sa date de naissance (3 entiers pour le jour, le mois et l'année). S'il s'agit du même jour et mois d'aujourd'hui (des constantes entières fixées à la date d'aujourd'hui), il affiche « Joyeux anniversaire Hamid ». Si le nom d'utilisateur n'est pas "hamid", il affiche "ERREUR : nom d'utilisateur inconnu".

Exercice 3.

Écrire un algorithme qui permet de lire une file de nombres entiers positifs (terminée par -1). Celui-ci affiche le nombre d'entiers pairs et le pourcentage par rapport au nombre d'entiers entrés.

Exercice 4.

Écrire un algorithme qui permet de vérifier si un nombre entier (entré par l'utilisateur) est un nombre premier ou non. Un nombre premier est un nombre qu'on peut diviser par lui-même et par le nombre 1 seulement.

Exercice 5.

Écrire un algorithme qui permet d'afficher les nombres premiers compris entre ~~2~~ et 100, ainsi que leur somme.

Exercice 6 :

- Écrire une fonction « reste (...) » qui permet de calculer le reste de la division entière de deux nombres entiers positifs reçus en paramètre.
- Utiliser cette fonction dans un algorithme qui permet de vérifier si un nombre est premier.
- Transformer cet algorithme en une fonction « estNombrePremier(...) » qui fait appel à la fonction reste(...).
- Utiliser cette fonction dans un algorithme qui permet d'afficher les 100 premiers nombres premiers.

Les fonctions série 1/2

Exercice 1:

Algô : Ex 1

Var : a, b, s, P : Réels

Début :

Ecrire ("a?") ; lire (a)

Ecrire ("b?") ; lire (b)

S \leftarrow a + bP \leftarrow a * b

Affiche Signe (S)

Affiche Signe (P)

Fin

fonction : AfficheSigne (x : réel) : rien

Début :

si ($x > 0$), alors :Ecrire (x , "est positif".)sinon si ($x < 0$), alorsEcrire (x , "est négatif".)

Sinon

Ecrire (x , "est nul.")

Finsi

Fin

Exercice 2:

Algô : Ex 2

Constant : NOM = "hamid"

J = 5

M = 10

Var : name ; chaîne de caractères

Début

Ecrire ("Votre nom ?")

Lire (name)

si correct (name, NOM), alors

ou si (name == NOM), alors

traitement (name, J, M)

sinon :

Ecrire ("ERREUR : nom incorrect.")

Finsi

Fin

fonction traitement (nom : chaîne ; j : entier,
m : entier) (mois?)

Var: jour, mois, an : entiers.

Début :

Ecrire ("Jour de naissance ?"); Lire ("Jour")

Ecrire ("Mois de naissance ?"); Lire ("mois")

Ecrire ("Année de naissance ?"); Lire ("an")

si (Jour = J) et (mois = M), alors

Ecrire ("Joyeux anniversaire", nom)

Fin si

Fin.

Exercice 3:

Algo : Ex3

Var: n, n, np : entier

p : réel.

Début :

Ecrire ("Saisir la file de nombres un par un
et terminer par -1").

$n \leftarrow 0$

$np \leftarrow 0$

répéter

Lire(n)

si ($n > 0$), alors

$n \leftarrow n + 1$

si ($n \bmod 2 = 0$) alors

$np \leftarrow np + 1$

Fin si

Fin si

jusqu'à ($n = -1$)

si ($n \neq 0$), alors

$p \leftarrow (np/n) * 100$

sinon

$p \leftarrow 0$

Fin si

Ecrire ("le nombre de pairs =", "np")

Ecrire ("Avec le pourcentage de", "p")

Fin.

Exercice 4:

fonction est premier (n : entier) : entier

Var: n, i : entier

Début :

Pour i allant de 2 à $\frac{n}{2} - 1$

si ($n \bmod i = 0$), alors

retourner (0)

Fin si

Fin pour

Retourner (1)

Fin

Algo : Ex 4

Var : n : entier

Début :

Ecrire ("n ?")

Lire (n)

si ("est premier (n) = 1"), alors

Ecrire (n, "est premier")

sinon,

Ecrire (n, "n'est pas premier")

Fin si

Fin

Exercice 5 :

Algo : Ex 5:

Var i : entier

S : entier

Début

$S \leftarrow 0$

Pour i allant de 2 jusqu'à 100

si ("est premier (i) = 1") alors

$S \leftarrow S + i$

Ecrire ('i, "

Fin si

Fin pour

Fin .

Exercice 6:

fonction Reste (a:entier, b:entier) : entier
 $a \% b$

Var r:entier.

Début

tq ($a > b$) faire

$a \leftarrow a - b$

Fin tq

Retourner (a)

Fin

Algo premier

Var a, i : entier

Début

Ecrire ("a?")

Lire (a)

$i \leftarrow 2$

premier \leftarrow vrai

tq ($i \leq a/2$) et (premier = vrai) faire

si (reste (a, i) = 0) alors

premier \leftarrow Faux

sinon

$i \leftarrow i + 1$

Fin si

Fin tq

si premier alors

Ecrire (a, "est premier")

sinon

Ecrire (a, "n'est pas premier")

Fin si

Fin

Algo Ex6 -4

Var c : entier // compteur
a : entier // v.t

Début

c \leftarrow 0

a \leftarrow 1

tq (c < 100) faire

si (est Nombre Premier (a) = Vrai)

Ecrire (a, "ln")

c \leftarrow c + 1

Fin si

a \leftarrow a + 1

Fin tq

Fin

Algorithmique II
Exercices de TD 1
Les Fonctions (Série 2/2)

Prof. Noureddine En-nahn

Exercice 1. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : une chaîne représentant un mot en minuscules (nom simple, verbe, adjectif) de la langue française.
- Résultat : le nombre de voyelles et le nombre de consonnes dans la chaîne.

Par exemple, le mot *élève* contient trois voyelles et deux consonnes.

Exercice 2. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : deux chaînes x et y représentant chacune un mot en majuscules (on n'utilise pas les accents) de la langue française.
- Résultat : *vrai* si x et y sont des anagrammes et *faux* sinon.

Rappelons que x est un *anagramme* de y si x est une permutation des lettres de y . Par exemple, le mot CHIEN est un anagramme du mot NICHE. De la même manière, le mot LIERRE est un anagramme du mot RELIER. Mais LIRE n'est pas un anagramme de LIERRE puisqu'il n'utilise qu'un seul R et qu'un seul E.

Exercice 3. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : une chaîne de caractères x représentant un mot ou une expression de la langue française dont tous les caractères sont en majuscules.
- Résultat : *vrai* si x est un palindrome et *faux* sinon.

Rappelons que x est un *palindrome* s'il se lit de la même manière de gauche à droite et de droite à gauche. Par exemple, le mot RADAR et l'expression MON NOM sont des palindromes. Mais l'expression LE SEL n'est pas (strictement) un palindrome à cause du caractère 'espace'.

Exercice 4. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : une phrase de la langue française ne contenant pas de symbole de ponctuation autre que le point final.
- Résultat : le nombre de mots dans la phrase.

Par exemple, la phrase "Le chat est couché dans le canapé." contient 7 mots.

Exercice 5. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : une phrase P de la langue française et un mot m , tous deux écrits en majuscules.
- Résultat : le nombre d'occurrences de m dans P .

Par exemple, la phrase "LE CHAT S'APPELLE PACHA." contient 2 fois le mot LE et 2 fois le mot CHA.

Exercice 6. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : une matrice M de dimension $n \times n$
- Résultat : *vrai* si M est la matrice identité et *faux* sinon.

Dans la matrice identité, tous les éléments de sa diagonale sont égaux à 1 et tous les autres éléments sont égaux à 0.

Exercice 7. Ecrire un algorithme permettant de résoudre le problème suivant :

- Données : deux matrices A et B de dimension $m \times n$
- Résultat : la somme $A + B$ des deux matrices.

Rappelons que si C est la somme de A et de B , alors pour toute rangée $1 \leq i \leq m$ et toute colonne $1 \leq j \leq n$ de C , nous avons $c_{ij} = a_{ij} + b_{ij}$.

Les fonctions série 2/2

Exercice 1 :

Algo Ex1

Var S = chaîne de caractère [1, Max]

nv, nc = entier

Début

Ecrire ("un mot ?")

Lire (S)

nv \leftarrow nb voyelles (S)

nc \leftarrow taille (S) - nv

Ecrire (S, "compte", nv, "voyelles et", nc, "consonnes")

Fin

fonction nb Voyelles (S: chaîne) : entier

Var c, i : entier

Début

c \leftarrow 0

Pour i allant de 1 à taille (S) faire

si (est voyelle (S[i])) alors

c \leftarrow c + 1

Fin si

Fin Pour

Retourner (c)

Fin

fonction est Voyelle (c: caractère) : booléen.

Var Voyelles : chaîne

i : entier

Début

Voyelles = "aeiouyàâèêëïïùûœ"

$i \leftarrow 1$
Pour i allant de 1 à taille (voyelles) faire
 si ($c = \text{voyelles}[i]$) alors
 Retourner (vrai)
 Fin si
Fin Pour
Retourner (Faux)

Fin

Exercice 2:

Algo : Ex2

Var ch1, ch2 : chaînes

Début

Ecrire ("Votre 1^e nom ?")

Lire (ch1)

Ecrire ("Votre 2^e nom ?")

Lire (ch2)

si est Anagramme (ch1, ch2) alors

Ecrire (ch1, "et", ch2, "sont anagrammes")

sinon

Ecrire (ch1, ch2, "ne sont pas anagrammes")

Fin si

Fin

Fonction estAnagramme (x : chaîne, y : chaîne) : booléen

Début

si taille (x) = taille (y) alors

 trier (ref x)

 trier (ref y)

 Retourner (est Egal (x, y))

sinon

Retourner (Faux)

Fin si
Fin

fonction trier (ref t : chaîne de caractère) . rien

Var i, j : entier

Début

Pour i allant de taille (n) jusqu'à 2 , pas = -1

Pour j allant de 1 à i - 1

si t[j] > t[j+1] alors

echanger (t, j, j+1)

Fin si

Fin Pour

Fin Pour

Fin

fonction echanger (ref t : chaîne , p1 : entier , p2 : entier)

Var vt : caractère

Début

vt \leftarrow t[p1]

t[p1] \leftarrow t[p2]

t[p2] \leftarrow vt

Fin

fonction Est égal (t1 : chaîne , t2 : chaîne) : booléen .

Var i : entier

Début :

Pour i allant de 1 à taille (chaîne) faire

si (t1[i] \neq t2[i]) alors

Retourner Faux

Fin si

Fin Pour
Retourner vrai

Fin

Exercice 3. Palindrome

Algo : Ex3

Var S : chaîne de caractères

Début

Ecrire ("Donner une chaîne")

Lire (S)

si (est Palindrome (S)) alors

Ecrire (S, "est palindrome")

sinon

Ecrire (S, "n'est pas palindrome")

Fin si

Fin

Fonction estPalindrome (S:chaîne) : booléen

Var i,j : entier

Début

i ← 1

j ← taille (S)

tq (i < j) faire

si (S[i] ≠ S[j]) alors

Retourner (Faux)

sinon

i ← i + 1

j ← j - 1

Fin tq

Retourner (vrai)

Fin

fonction est Palin (ref s : chaîne, deb : entier, fin : entier) : booléen
Début

si (deb > fin) alors

retourner (vrai)

sinon si (s[deb] ≠ s[fin]) alors

retourner (Faux)

sinon

retourner Palin (s, deb+1, fin - 1)

Fin si

Fini;

Fin

Exercice 6 :

Algo Ex6 - 2

Var M : tableau de réels [1...Max][1...Max]

n : entier

Début

lire Mat (ref M, ref n)

si (est Id (M, n)) alors

Ecrire ("la matrice est une identité")

sinon

Ecrire ("NON")

Fin si

Fin

fct lire Mat (ref M : tab [1...Max][1...Max], ref n : entier) : vide

Var i, j : entier

Début

Ecrire ("donner n ?")

Lire (ref n)

Pour i allant de 1 à n faire

Pour j allant de 1 à n faire
Ecrire (" $M[i, i, " ", j, "] = "$)
Lire (ref $M[i][j]$)

Fin Pour j

Fin Pour i

Fin

fonction est Id (M : tab d'entiers [1...Max][1...Max], n : entier),
booléen.

Var i, j : entiers

Début

Pour i allant de 1 à n faire
si $M[i][i] \neq 1$ alors
Retourner (Faux)

Fin si

Fin Pour

Pour i allant de 2 à n faire

Pour j allant de 1 à $(i-1)$ faire
si $M[i][j] \neq 0$ alors
Retourner (Faux)

Fin si

Fin Pour j

Pour j allant de $i+1$ à n faire

si $M[i][j] \neq 0$ alors
Retourner (Faux)

Fin si

Fin Pour j

Fin Pour i

Retourner (Vrai)

Fin

fonction vérifierO (M : tab de réels [1..Max][1..Max], i ,
 $cdeb, cfin$: entiers) : booléen.

Var j : entier

Début

Pour j allant de $cdeb$ à $cfin$ faire
si $M[i][j] \neq 0$ alors

Retourner (Faux)

Fin si

Fin Pour

Retourner (Vrai)

Fin

fonction Id2 (M : t[1...Max][1..Max], n : entier) : booléen

Var i : entier

Début

Pour i allant de 1 à n faire

si ($M[i][j] \neq 1$) alors

Retourner (Faux)

Fin si

si (VérifierO ($M, i, 1, i-1$) = Faux) alors

Retourner (Faux)

sinon

si (VérifierO ($M, i, i+1, n$) = Faux) alors

Retourner (Faux)

Fin si

Fin si

Fin Pour

Retourner (Vrai)

Fin

fct est $\text{Id3}(n \dots, n \dots)$: booleen

Var i, j : entiers

Début

Pour i allant de 1 à n faire

Pour j allant de 1 à n faire

si ($i = j$) alors

si ($M[i][j] \neq 1$) alors

Retourner (Faux)

Fin si

sinon ($i \neq j$)

si ($M[i][j] \neq 0$) alors

Retourner (Faux)

Fin si

Fins:

Fin Pour j

Fin Pour i

Retourner (Vrai)

Fin

Exercice 7.

Algo Ex 7_2

Var A, B : tableaux de réels $[1 \dots \text{Max}] [1 \dots \text{Max}]$

Var n, m : entiers

Début

Ecrire ("n?")

Lire (n)

Ecrire ("m?")

Lire (m)

Lire Mat (A, m, n)

Lire Mat (B, m, n)

Somme Mat (A, B, C, m, n)

Affiche Mat (c, m, n)

Fin

jet Lire Mat (A : tableau, n : entier, m : entier)
Var : i, j : entiers.

Début

Pour i allant de 1 jusqu'à m faire

Pour j allant de 1 jusqu'à n faire

Ecrire ("Donner A[", i, ",", ", j] = ")

Lire (A[i][j])

Fin Pour j

Fin Pour i

Fin

jet Somme Mat (A, B, C : tableaux de réels [][], m : entier, n : entier)

Var : i, j : entiers

Début

Pour i allant de 1 jusqu'à m faire

Pour j allant de 1 jusqu'à n faire

C[i][j] \leftarrow A[i][j] + B[i][j]

Fin pour j

Fin Pour i

Fin

Exercice 8:

Algo Ex8 - 2 :

Var A, B, C : tableaux de réels [1..Max][1..Max]
l, n, m : entiers

Début :

Ecrire (" le nombre de lignes de A ")

Lire (&m)

Ecrire (" le nombre de colonnes de A = le nombre
de lignes de B")

Lire (&n)

Ecrire (" le nombre de colonnes de B ")

Lire (&l)

Lire Mat(A, m, n)

Lire Mat(B, n, l)

ProdMat(A, B, C, m, n, l)

AfficheMat(C, m, l)

Fin

Def ProdMat(A, B : tableaux de réels [][] , m, n, l : entiers)
Var i, j, k : entiers.

Som : réel

Début

Pour i allant de 1 à m faire

Pour j allant de 1 à l faire .

Som \leftarrow 0

Pour k allant de 1 à n faire

Som \leftarrow Som + A[i][k] + B[k][j]

Fin Pour k

Fin Pour j

Fin Pour i

Fin

Série n°2 (Fonctions 2)

Exercice 1: Minimum et Somme d'un tableau.

a - fonction recursive qui détermine la valeur min dans un tableau d'entrée de n éléments [0...Max]:

fct Min_iter (iteratif) (T: tableau d'entier [0...Max], n; entier) : entier

Début

m ← T[0]

i ← 1

tq (i < n) faire

 si (T[i] < m) alors

 m ← T[i]

 fin si

 i ← i + 1

fin tq

retourner (m).

si (i > n) alors

 retourner (m)

sinon

 si (T[i] < m) alors

 m ← T[i]

 fin si

 retourner min_rec (T, n, i+1, m).

fin si

fin .

Algô-principal:

min ← min_rec (T, n, 1, T[0])

b - fonction renvoie la valeur de la somme des éléments du tableau:

fct Somme (tableau T[1...Max] : entier, S : entier, n : entier, i : entier) : entier,

Début

si ($i > n$) alors
retourner (S)

sinon

S = S + T[i]

retourner Somme (T, S, n, i+1)

fin si

fin.

Algo principal :

resultat \leftarrow Somme (T, S, n, 0)

Exercice 2.

fonction PGCD (a:entier, b:entier) : entier.

Début

si ($a \% b = 0$) alors

retourner (a)

si ($a > b$) alors

a = a - b

retourner PGCD (a, b) } Retourner PGCD (a-b, b)

fin si

si ($a < b$) alors

b = b - a

retourner PGCD (a, b)

fin si

fin

fonction PGCD (a : entier, b : entier) : entier.

Début

si ($a \cdot b = 0$) alors

retourner (b)

sinon

si ($a > b$) alors

retourner ($b, a \% b$)

finsi

si ($a < b$) alors

| retourner PGCD (b, a)

finsi

fins:

fin

Exercice 3:

fonction ExistCar (c : caractère, x : chaîne, i : entier) :
booléen.

Début :

si ($i \geq \text{taille}(x)$) alors

retourner (Faux).

sinon

si ($x[i] = c$) alors

| retourner (vrai)

sinon

| retourner ExistCar ($c, x, i+1$)

finsi

finsi

fin

Algo - principal

{ ExistCar ($c, x, 0$) }