Exercice 1 :

résolution d'une équation du second degré

exercice 2 :

Ecrire un algorithme qui lit un entier positif n puis affiche tous ses diviseurs.

Exercice 3 :

Ecrire un algorithme qui lit un entier positif n puis calcule et affiche son factoriel selon la formule n! = 1 ´2 ´… ´n.

Exercice 4 :

Ecrire un algorithme qui lit un entier positif et vérifie si ce nombre est premier ou non

Exercice 5 :

Ecrire un algorithme qui lit une chaîne de caractères puis affiche son inverse.

Exemple : Si la chaîne entrée est ”algo”, l'algorithme doit afficher ”ogla”

Exercice 5.1 : affichage de l'inverse d'une chaîne de caractères

Algorithmeinverse

Variables

i, L : Entier

ch1, ch2 : Chaîne

Début

Ecrire(”Entrer une chaîne : ”) Lire(ch1)

L ¬Long(ch1)

ch2 ¬””

Pouri de1 àL Faire

ch2 ¬ch1[i] + ch2

FinPour

Ecrire(”Inverse de la chaîne = ”,ch2)

Fin

Exercice 6 :

Ecrire un algorithme qui permet de compter le nombre de mots dans une phrase.

La phrase commence obligatoirement par une lettre et les mots sont séparés par des espaces

AlgorithmeComptage_Mots

Variables

i, L, nb_mot : Entier

phrase : Chaîne

Début

Ecrire(”Entrer une phrase non vide : ”) Lire(phrase)

L ¬Long(phrase)

nb_mot ¬1

Pouri de1 àL Faire

Si (phrase[i] = ” ”) Alors

nb_mot ¬nb_mot + 1

FinSi

FinPour

Ecrire(”Nombre de mots = ”,nb_mot)

Fin

Exercice 6 :

Ecrire un algorithme « Palind » qui lit une chaîne de caractères et vérifie si cette chaîne est un palindrome ou non

AlgorithmePalind

Variables

ch : Chaîne

i, L : Entier

Pal : Booléen

Début

Ecrire(”Entrer une chaîne non vide : ”) Lire(ch)

L ¬long(ch)

Pal ¬Vrai

i ¬1

TantQue(i <= L/2) et (Pal) Faire

Si(ch[i] = ch[L-i+1]) Alors

i ¬i + 1

Sinon

Pal ¬Faux

FinSi

FinTQ

Si(Pal) Alors

Ecrire(ch, ” est un palindrome”) Sinon

Ecrire(ch, ” n’est pas un palindrome”) FinSi

Fin

Exercice 7:

1. On appelle bigrammeune suite de deux lettres. Ecrire une procédure qui calcule le nombre d’occurrences d’un bigramme dans une chaîne de caractères.

2. Peut-on transformer cette procédure en fonction ? si oui écrire cette fonction.

Procédurefréqunece(bigram:Chaîne[2] ; chn :Chaîne ; Var nb : Entier)

Variables

i, L : Entier

Début

L ¬Long(chn)

nb ¬0

Pour i de 1à (L-1) Faire

Si (chn[i]=bigram[1]) ET (chn[i+1]=bigram[2])

Alors

nb ¬nb + 1

FinSi

FinPour

Fin

Cette procédure possède un seul paramètre résultat de type entier, donc

elle peut être remplacée par une fonction.

Fonctionfréqunece(bigram:Chaîne[2] ; chn :Chaîne) : Entier

Variables

i, L : Entier

Début

L ¬Long(chn)

nb ¬0

Pour i de 1à (L-1) Faire

Si (chn[i]=bigram[1]) et (chn[i+1]=bigram[2])

Alors

nb ¬nb + 1

FinSi

FinPour

fréquence ¬nb

Fin

Exercice 8 :

Soient M1 et M2 deux matrices à n lignes et m colonnes. On veut écrire une procédure qui calcule les éléments de la matrice M3=M1+M2. (produit de deux matrice)

Procédure SomMat(M1, M2 : Mat ; Var M3 : Mat)

Variables

i, j : Entier

Début

Pouri de1 àn Faire

Pour jde 1à mFaire

M3[i,j] ¬M1[i,j]+ M2[i,j]

FinPour

FinPour

Fin

Solution produit de deux matrice :

Procédure ProdMat(M1 : Mat1; M2 : Mat2; Var M3 : Mat3)

Variables

i, j, k : Entier

Début

Pouri de1 àn Faire

Pour jde 1à pFaire

M3[i,j] ¬0

Pourk de1 àm Faire

M3[i,j]¬M3[i,j]+M1[i,k]*M2[k,j]

FinPour

FinPour

FinPour

Fin

Exercice 10 :

Ecrire un algorithme permettant de trouver tous les nombres premiers inférieurs à 400 et qui utilise la méthode suivante :

· Créer un tableau T pouvant contenir 400 entiers

· Initialiser chaque élément du tableau à son indice c’est-à-dire T[1]=1 ; T[2]=2 ; … T[400]=400

· Remplacer tous les multiples de 2 par 0 sauf 2

· Chercher le prochain élément différent de 0 dans le tableau c’est à dire T[3] et remplacer tous les multiples de 3 par 0 sauf 3

· Continuer ce processus jusqu’à avoir T[i] ³20 (20 étant la racine carrée de 400)

· Afficher tous les éléments non nuls de T.

Solution Algorithme Premiers

Constantes

n = 400

Types

Tab = Tableau[1..n] de Entier

Variables

Prem : Tab

i, j : Entier

Procédure initialiser(Var T : Tab)

Début

Pour i de1 àn Faire

T[i] ¬i

FinPour

Fin

Procédure mise_a_zero(Var T : Tab)

Début

Pour i de1 à20 Faire

Si(T[i] # 0) Alors

Pour j dei+1 àn Faire

Si(T[j] Mod T[i] = 0) Alors

T[j] ¬0

FinSi

FinPour

FinSi

FinPour

Fin

Procédure afficher(T : Tab)

Début

Pour i de1 àn Faire

Si(T[i] # 0) Alors

Ecrire(T[i])

FinSi

FinPour

Fin

Début

initialiser(Prem)

mise_a_zero(Prem)

afficher(Perm)

Fin.