Université Hassan 1er

Faculté des Sciences et Techniques de Settat

PROGRAMMATION EN «C» ET STRUCTURES DE DONNÉES TRAVAUX DIRIGÉS

Professeur Laachfoubi Nabil

Département des Mathématiques et Informatique

Sommaire

SERIE 1 – Les Tests Logiques, Les Procédures et Les Fonctions 3
SERIE 2 – Les Boucles et La Récursivité 4
SERIE 3 – Les Tableaux.....6
SERIE 4 – Les Tableaux (suite) 7
SERIE 5 – Les Chaînes de Caractères 8

SERIE 1 – Les Tests Logiques, Les Procédures et Les Fonctions

1. Soient les procédures suivantes :

Algorithme Procedure P1( ) **Var** A, B, C **: Entier ;** **Debut** **Lire (**A , B**)** ; C ← A ; A ← B ; B ← C ; **Afficher (**A , B**)** ; **Fin**	**Algorithme Procedure P2( )** **Var** A, B **: Entier ;** **Debut** **Lire (**A , B**)** ; A ← A + B ; B ← A - B ; A ← A - B ; **Afficher (**A , B**)** ; **Fin**

   a) Remplissez les cellules grisées sur les tableaux de valeurs ci-dessous.
   b) Quel rôle joue chacune des procédures ?
   c) Laquelle de ces procédures est la plus rapide ?

Procédure P1				
Entrée			**Sortie**	
A	**B**		**A**	**B**
0	2	→		
-1	1			
4	2			

Procédure P2				
Entrée			**Sortie**	
A	**B**		**A**	**B**
0	2	→		
-1	1			
4	2			

2. Ecrire une procédure permettant d'afficher la valeur max entre deux entiers qu'un utilisateur saisit au clavier.

3. Ecrire une fonction permettant de retourner la valeur max entre deux entiers qu'un utilisateur saisit au clavier.

4. Ecrire une procédure prenant en paramètre deux variables de type entier et permettant d'afficher la valeur max entre ces deux variables.

5. Ecrire une fonction prenant en paramètre deux variables de type entier et permettant de retourner la valeur max entre ces deux variables.

6. Refaire les exercices 2, 3, 4 et 5 mais cette fois-ci en prenant en compte trois valeurs entières.

SERIE 2 – Les Boucles et La Récursivité

7. Soit la fonction f(n) = n! avec (0! = 1 et n! = n*(n-1)! Pour tout n > 0)
   a) Ecrire une procédure sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et d'afficher la valeur de f(n).
   b) Ecrire une procédure avec comme paramètre la variable n, permettant d'afficher la valeur de f(n).
   c) Ecrire une fonction sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et de retourner la valeur de f(n).
   d) Ecrire une fonction avec comme paramètre la variable n, permettant de retourner la valeur de f(n).
   e) Ecrire une procédure avec comme paramètres les variables n et fact, permettant de calculer la valeur de f(n) et stocker le résultat dans l'adresse de fact.
   f) Ecrire une procédure, permettant de trouver la plus petite valeur de n ainsi que celle de f(n) pour laquelle f(n) >= 121.
   g) Ecrire une fonction récursive avec comme paramètre la variable n et permettant de retourner la valeur de f(n).

8. Soit la suite U(n) = 3*U(n - 1) + 2 pour tout n > 0 avec U(0) = 1
   a. Ecrire une procédure sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et d'afficher la valeur de U(n).
   b. Ecrire une procédure avec comme paramètre la variable n, permettant d'afficher la valeur de U(n).
   c. Ecrire une fonction sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et de retourner la valeur de U(n).
   d. Ecrire une fonction avec comme paramètre la variable n, permettant de retourner la valeur de U(n).
   e. Ecrire une procédure, permettant de trouver la plus petite valeur de n ainsi que celle de U(n) pour laquelle U(n) >= 121.
   f. Ecrire une fonction récursive avec comme paramètre la variable n et permettant de retourner la valeur de U(n).
   g. Ecrire une procédure récursive avec comme paramètres la variable n et une autre variable passée par adresse et permettant de retourner la valeur de U(n).

9. Soit la série suivante :

$$S(n) = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \quad avec\ n\ entier > 0$$

   a. Ecrire une procédure sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et d'afficher la valeur de S(n).
   b. Ecrire une procédure avec comme paramètre la variable n, permettant d'afficher la valeur de S(n).
   c. Ecrire une fonction sans paramètre, permettant de demander à l'utilisateur de saisir la valeur de n et de retourner la valeur de S(n).
   d. Ecrire une fonction avec comme paramètre la variable n, permettant de retourner la valeur de S(n).
   e. Ecrire une procédure, permettant de trouver la plus petite valeur de n ainsi que celle de S(n) pour laquelle S(n) >= 3.
   f. Ecrire une fonction récursive avec comme paramètre la variable n et permettant de retourner la valeur de S(n).
   g. Ecrire une procédure récursive avec comme paramètres la variable n et une autre variable passée par adresse et permettant de retourner la valeur de S(n).

10. Ecrire une procédure permettant de résoudre l'équation de second degré $f(x) = a.x^2 + b.x + c = 0$

11. Ecrire une procédure permettant de résoudre le système d'équations suivant :

    $a.x + b.y = c$

    $d.x + e.y = f$

SERIE 3 – Les Tableaux

12. Soit V un vecteur monodimensionnel de 5 entiers, écrire les procédures suivantes :
    a. Procédure sans paramètre permettant de calculer la somme des 5 entiers,
    b. Procédure sans paramètre permettant de calculer le produit des 5 entiers,
    c. Procédure sans paramètre permettant de calculer la moyenne des 5 entiers,
    d. Procédure sans paramètre permettant de trouver la valeur max entre les 5 entiers,
    e. Procédure sans paramètre permettant de trouver la valeur min entre les 5 entiers,
    f. Procédure sans paramètre permettant de calculer la somme, le produit, la moyenne, le max et le min des 5 entiers.
    g. Procédure sans paramètre permettant de trier les 5 entiers dans l'ordre croissant
    h. Procédure sans paramètre permettant de trier les 5 entiers dans l'ordre décroissant.

13. Ecrire une fonction prenant en paramètre un vecteur V, sa taille et une position p. Cet algorithme doit vous permettre de retourner l'indice de la valeur min du vecteur en partant de la position p.

14. Ecrire une procédure sans paramètre, permettant de trier un vecteur d'entiers en s'appuyant sur la fonction de la question 14.

15. Soient U et V deux vecteurs d'entiers, lu et lv leurs tailles respectives.
    a. On suppose que U et V sont déjà triés, écrire une procédure sans paramètre, permettant de fusionner ces deux vecteurs pour obtenir un troisième vecteur trié.
    b. Ecrire une fonction permettant de fusionner les deux vecteurs en un seul.
       Prototype : int * FusionVecteurs(int *U, int lu, int *V, int lv)
    c. On suppose que les deux vecteurs U et V sont déjà triés, écrire une fonction permettant de fusionner les deux vecteurs en un seul en effectuant un remplissage du nouveau vecteur dans un ordre croissant. (Remarque : à la fin du remplissage du nouveau vecteur, celui-ci va se retrouver trié également)
       Prototype : int * FusionCroissanteVecteurs(int *U, int lu, int *V, int lv)

SERIE 4 – Les Tableaux (suite)

16. Ecrire une procédure permettant de demander à l'utilisateur de fournir la taille d'un vecteur d'entiers, puis de saisir les valeurs pour ce vecteur avant d'afficher la liste de ces valeurs.

17. Ecrire une fonction permettant en paramètre la taille d'un vecteur d'entiers et retournant l'adresse mémoire de ce vecteur.

18. Ecrire une procédure prenant en paramètre l'adresse d'un vecteur d'entiers et sa taille, et permettant de saisir les éléments pour ce vecteur.

19. Ecrire une procédure prenant en paramètre l'adresse d'un vecteur d'entiers et sa taille, et permettant d'afficher à l'écran les éléments de ce vecteur.

20. Ecrire une procédure permettant de calculer la somme de deux matrices de taille 2x3.

21. Ecrire une procédure permettant de calculer le produit de deux matrices de tailles 4x3 pour T et 3x2 pour V.

SERIE 5 – Les Chaînes de Caractères

22. Trouver dans un tableau de caractères toutes les sous-chaînes de caractères de longueur maximale dont la première lettre et la seconde sont identiques. Affichez le nombre de sous-chaînes trouvées.

23. Trouver dans un tableau de caractères toutes les sous-chaînes de caractères dont la première lettre et la seconde sont identiques. Affichez le nombre de sous-chaînes trouvées.

24. Ecrire la fonction « **PositionChaine »** permettant de rechercher la position de la première occurrence d'une chaine « **t** » à l'intérieure d'une autre chaine « **s** ». Si la chaine « **t** » n'existe pas à l'intérieur de la chaine « **s** », la fonction renvoie la valeur **-1**.
    Prototype : **int** PositionChaine (**char** ***** s , **char *** t)

25. Ecrire la fonction « **PositionChaineIndice »** permettant de rechercher la première position d'une chaine « **t** » à l'intérieure d'une autre chaine « **s** » à partir d'un indice donné. Si la chaine « **t** » n'existe pas à l'intérieur de la chaine « **s** » à partir de cet indice, la fonction renvoie la valeur -**1**.
    Prototype : **int** PositionChaineIndice (**char** ***** s , **char *** t , **unsigned int** pos)

26. Ecrire une procédure « **PositionsChaine »** permettant de rechercher et d'afficher toutes les positions d'une chaine « **t** » à l'intérieure d'une autre chaine « **s** ».
    Prototype : **void** PositionsChaine (**char** ***** s , **char *** t)

27. Ecrire la fonction « ListePositionsChaine » permettant retourner un tableau contenant toutes les positions des occurrences de la chaine « t » à l'intérieure de la chaine « s ».
    Prototype : **int *** ListePositionsChaine (**char *** s , **char *** t , **int *** nb_occur)

28. Adapter la fonction « ListePositionsChaine » (exo 28) pour retourner un tableau contenant toutes les positions du mot « t » ou de l'occurrence « t » à l'intérieure de la chaine « s ».
    Prototype : **int *** ListePositionsChaine (**char *** s , **char *** t , **int *** nb_occur , **char** opt)

29. Ecrire la fonction « SaisirChaine » permettant de saisir une chaine de caractères au clavier et lui allouant la taille mémoire exacte sans pour autant connaitre sa taille à l'avance.
    Prototype : **char *** SaisirChaine ()