Lycée Garçons Sousse

Novembre 2022

Prof. Zied Fridhi

Série d'exercices n°9

Algorithmique et Programmation

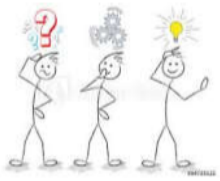
4ème Sciences de l'Informatique

Durée : **H ** Coef : 3**

Exercice n°1 :

1. Soit l'algorithme, de la fonction inconnue, suivant :

```
Fonction Inconnue( F : texte ) : .....
DEBUT
        ouvrir("eleve.txt", F, "r" )
        Retourner Fin_Fichier(F)
FIN
```



Après exécution de cet algorithme, et pour chacune des propositions suivantes, mettre dans chaque case la lettre **V** si la réponse est correcte ou la lettre **F** dans le cas contraire.

☐ Le fichier F ne contient pas des données, si la fonction Inconnue retourne Vrai.

☐ Le fichier F contient des données, si la fonction Inconnue retourne le numéro de la ligne, sinon retourne **-1**.

2. Soient les différentes algorithmes de la procédure « afficher », ci-dessous (avec N un entier strictement supérieurs à 1).

N° de l'algorithme	L'Algorithme
1	procedure afficher (N : entier) DEBUT Si N=1 ALORS ECRIRE (N) SINON afficher(N-1) ECRIRE (N) FINSI FIN
2	procedure afficher (N : entier) DEBUT Si N=1 ALORS ECRIRE (N) SINON ECRIRE (N) afficher(N-1) FINSI FIN
3	Procedure afficher (N : entier) DEBUT Si N=1 ALORS ECRIRE (N) SINON afficher(N-1) FINSI FIN

Pour chacun des résultats ci-dessous, mettre dans chaque case le numéro de l'algorithme de la procédure « afficher » qui correspond à l'affichage de ces résultats :

☐ 1 ☐ de 1 à N ☐ boucle infinie ☐ de N à 1

Exercice n°2 :

Soit à remplir au hasard une matrice carrée d'ordre n (3< n < 23) par les entiers 0 ou 1.

NB : la fonction **alea(a,b)** permet de générer un entier aléatoire appartenant à l'intervalle [a, b-1].

A partir de cette matrice, on désire créer le programme suivant qui permet de :

- ✓ Remplir un fichier **FH** stocké physiquement sous la racine "c:\" par le nom "convert.dat". On remplit le fichier par des enregistrements renfermant les champs suivant :
  - ❖ **Hex** : représente la conversion en hexadécimale de chaque ligne de la matrice.
  - ❖ **Oct** : représente la conversion en octale de chaque colonne de la matrice.
  - ❖ **Dec** : représente la somme de **Hex** et **Oct**
  - ❖ **Nb** : représente le nombre de lettres existant dans la valeur de champs **Hex**.
- ✓ Trier le fichier dans l'ordre croissant par le nombre de lettre existante.
- ✓ Rechercher **si** le fichier admet le nombre X de lettre et afficher le premier enregistrement rencontré ainsi que sa **position** correspondante dans le fichier, avec X étant une valeur saisie par l'utilisateur **sinon** afficher le message suivant « **le nombre de lettre cherché est inexistant** »

Exemple : Pour n=9

M

	1	2	3	4	5	6	7	8	9
1	1	1	1	0	0	1	1	1	0
2	1	0	0	0	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
4	0	0	1	1	0	0	0	1	0
5	1	1	0	0	1	1	0	0	1
6	0	1	1	1	0	1	1	1	1
7	0	0	1	0	0	0	0	1	0
8	1	1	1	1	1	1	0	1	0
9	1	1	1	0	0	0	1	1	1

Avant le tri FH

1CE	723	929	2
11F	533	634	1
1FF	557	878	2
62	152	204	0
199	322	619	0
EF	732	713	2
42	711	523	0
1FA	757	1001	2
1C7	311	656	1

Après le tri de FH

62	152	204	0
199	322	619	0
42	711	523	0
11F	533	634	1
1C7	311	656	1
1CE	723	929	2
1FF	557	878	2
EF	732	713	2
1FA	757	1001	2

- La première ligne de la matrice contient **111001110** son équivalent en hexadécimale est **ICE**.
- La première colonne de la matrice contient **111010011** son équivalent en octale est **723.**
- La somme de la première ligne et la première colonne de la matrice M (l'équivalant de chacune en décimale) est : 462+467 = 929.
- La valeur de NB =2 puisque ICE admet de 2 Lettre.

Travail demandé :

1- Ecrire l'algorithme du problème en le décomposant en modules.

2- Ecrire les algorithmes des modules envisagés.