Université Ibn Tofail

Ecole Nationale des Sciences Appliquées

Kenitra

Année Universitaire

2015-2016

Contrôle Continu (Durée : 02h)

Exercice 1 : (5 pt)

Soit T un tableau contenant N entiers. On propose d'écrire un algorithme qui permet d'éclater T en deux tableaux : **TN** (contenant les éléments **négatifs** de T) et **TP** (contenant les éléments **positifs** de T) et les afficher.

Exercice 2 : (3 pt)

Une année est dite bissextile (elle aura 366 jours) :

1. si l'année est divisible par 4 et non divisible par 100, **ou**
2. si l'année est divisible par 400.

Sinon, l'année n'est pas bissextile (elle a 365 jours).

Question : Ecrire un algorithme qui affiche la liste des années bissextiles comprises entre 2000 et 2100

Exercice 3 : (4pt)

1. Ecrire un algorithme qui calcule la valeur approchée de π en utilisant la formule ci-dessous. Sachant que le calcul s'arrête lorsque le terme $\frac{1}{x}$ est plus petit que ε (donné par l'utilisateur).

$$\frac{\pi}{4} \simeq 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \cdots$$

2. Ecrire un algorithme qui donne une approximation de e^x pour n donné, sachant que :

$$e^x \cong 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Exercice 4: (8pt)

L'exercice s'intéresse à la manipulation des matrices carrées (le programme principal n'est pas demandé)

1. Ecrire une procedure « **SAISIE** » qui permet de sasir les éléments d'une matrice carrée **M** d'ordre **n**
2. Ecrire une procedure « **AFFICHAGE** » qui permet d'afficher les éléments d'une matrice carrée **M** d'ordre **n**
3. Ecrire une procedure « **TRANSPOSEE** » qui permet de retourner la transposée d'une matrice M d'ordre n. sachant que L'opération de transposition consiste à inverser les lignes et les colonnes en effectuant une symétrie par rapport à la diagonale principale de la matrice.

Exemple : La matrice devient

1	2	3
4	5	6
7	8	9

1	4	7
2	5	8
3	6	9

4. Ecrire une fonction « **TRACE** » qui calcule la trace d'une matrice carrée M d'ordre n sachant que :

$$\text{Trace(M)} = \sum_{i=1}^{n} M_{ii}$$

5. Ecrire une foction « **NOMBRE_NUL** » qui calcule le nombre d'élements nuls dans la matrice

Pr : I.OUMAIRA

Université Ibn Tofail

Ecole Nationale des Sciences Appliquées Kenitra

Année Universitaire 2015-2016

Contrôle Terminal (Durée : 02h)

Exercice 1 : (4 pt)

Voir la feuille de réponses

Exercice 2 : (5pt)

Le tri par insertion consiste à sélectionner un élément du tableau et à l'insérer directement à la bonne position dans la partie du tableau déjà triée. On procède en trois étapes :

1. On place l'élément à trier dans une variable temporaire.
2. Tant que les éléments du tableau qui précèdent l'élément à trier lui sont supérieurs, on décale ces éléments d'une position en récupérant l'espace vide laissé par l'élément à trier.
3. On insère ensuite la variable temporaire à la nouvelle position laissée vacante par le décalage.

Questions :

1. Ecrire une fonction **remplir_list** permettant de créer une liste l, de la remplir d'une manière automatique et aléatoire de n valeurs et la retourner.
2. Ecrire une fonction **tri_insertion** permettant de trier une liste l en utilisant le tri par insertion
3. Ecrire un programme principal afin de tester les fonctions définies

Exercice 3 : (3pt)

Ecrire un programme qui effectuera les tâches suivantes :

1. Demander à l'utilisateur de saisir une phrase
2. Compter les **mots** de cette **phrase**, puis afficher ce nombre de mots.

- Ne pas compter les espaces au début et à la fin de la phrase
- On considérera que le séparateur des mots est le caractère 'espace' et que l'utilisateur ne met pas plus de 1 espace entre chaque mot

```
Entrer une chaine : Bonjour à toutes et à tous
la phrase contient  6  Mots
>>>
```

Figure 1 : Exemple d'exécution

Pr : I.OUMAIRA

Exercice 4: (3pt)

Écrire un programme Python permettant de lire un fichier texte "index.html", et d'en afficher uniquement les lignes qui contiennent le caractère "@" afin de rechercher les adresses email.

Exercice 5 : (5pt)

Ecrire un programme en python, pour réaliser le jeu suivant :

Le joueur (qui est l'utilisateur) doit trouver, en moins de 10 essais, un nombre compris entre 1 (inclus) et 100 (inclus), généré aléatoirement par l'ordinateur. On stockera ce nombre aléatoire dans une variable nommée sa, la valeur saisie par l'utilisateur sera stockée dans une variable nommée nb.

A chaque essai du joueur, on lui indique le numéro de son essai, le nombre d'essais restants, et si le nombre qu'il a donné est plus grand ou plus petit que le nombre à trouver.

```
---------------------------------------------------
c'est votre essai N° :  1
Entrer un nombre : 23
le nombre est plus petit que la valeur recherchée -
il vous reste 9 essai
---------------------------------------------------
c'est votre essai N° :  2
Entrer un nombre : 55
le nombre est plus grand que la valeur recherchée
il vous reste 8 essai
---------------------------------------------------
c'est votre essai N° :  3
Entrer un nombre : 46
c'est gangé
```

Figure 2 : Exemple d'exécution

Pr : I.OUMAIRA