|  |  |
| --- | --- |
| Résultat de recherche d'images pour "ofppt logo" | ***Office de la Formation Professionnelle*** ***et de la Promotion du Travail*** |

**DRTTA - ISMO NTIC – TETOUAN**

|  |
| --- |
| **Examen de Fin de Module**  **Au titre de l’année 2022/2023**  *M02 : Les bases d’algorithmique* |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Filière*** | *Développement Digital* | ***Épreuve*** | *Théorique* |
| ***Année*** | *1 er Année* | ***Barème*** | */40* |
| ***Niveau*** | *Technicien Spécialisé* | ***Durée*** | *2h30*  ***Variante****:1* |

**PARTIE I : ALGORITHMIQUE (12 PTS)**

**EXERCICE 1 (4 PTS)**

On représente un brin d’ADN par une chaîne de caractères qui peut contenir quatre caractères différents : ’A’ (Adénine), ’C’ (Cytosine), ’G’ (Guanine) et ’T’ (Thymine).

— Écrire une fonction **estADN(Mot : chaine des caracteres ) : booleen**  qui prend en argument une chaîne de caractères et renvoie Vrai si cette chaîne correspond à un brin d’ADN et qui renvoie Faux sinon.

***Par exemple :*** *("TTGAC") et estADN("GCAATAG") renvoient Vrai mais estADN("AMOG") et estADN("CaTg") renvoient Faux.*

**EXERCICE 2 (8 PTS)**

1. Ecrire une fonction récursive qui permet de calculer la suite suivante  (4 pts) :
2. Ecrire une procédure **Dessiner ( N : entier )** qui permet de dessiner la forme suivante en utilisant les termes de la suite définit dans la question 1

(4 pts )

*Exemple : pour N=5*

U1

U2 U2

U3 U3

U4 U4

U5 U5

**PARTIE II : PYTHON (28 PTS)**

**EXERCICE 1 (4 PTS)**

Écrire une fonction **Somme(chaine : string )** qui prend en argument une chaîne de caractères comprenant des entiers séparés par des symboles +, comme "7+52+3" ou "2+6+74+13" et qui renvoie le résultat de la somme. Si l’argument n’a pas le bon format, somme doit renvoyer -1. Contrat: Par exemple, somme("3+8") renvoie 11 mais somme("+7+8") et somme("4+3+9+") renvoient -1

**EXERCICE 2 `(4 PTS)**

Un coefficient binomial est défini pour deux entiers positifs n et k tels que n≥k. C’est le nombre de parties de k éléments dans un ensemble de n éléments. On le note : , et sa valeur est calculée par la formule suivante :

1. Écrire la fonction fact(p) qui reçoit en paramètre un entier positif p, et qui retourne la valeur de factorielle p : p! = 1 \* 2 \* 3 \* … \* (p-1) \* p. NB : La fonction fact(0) retourne 1 (1 pt )
2. Écrire la fonction **produit(n,k)** qui reçoit en paramètres deux entiers positifs n et k tels que n≥k, et qui retourne la valeur du produit : n \* (n-1) \* (n-2) \* … \* (n-(k-1)) (1 pt)
3. Écrire la fonction **binomial(n,k)** qui reçoit en paramètres deux entiers positifs n et k tels que n≥k, et qui retourne la valeur du coefficient binomial . (1 pt)

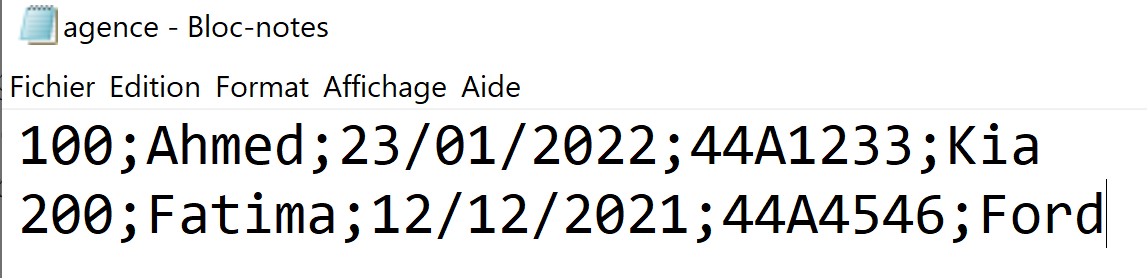
*Exemple : La fonction binomial (6, 3) retourne le nombre 20*

1. Écrire la fonction **liste\_binomiaux(n**) qui reçoit en paramètre un entier positif n, et qui retourne la liste des coefficients binomiaux tel que : k = 0, 1, 2, 3, …, n (1 pt)

*Exemple : La fonction liste\_binomiaux (6) => la liste [1, 6, 15, 20, 15, 6, 1 ]*

**EXERCICE 3 (20 PTS)**

Une agence de location des voitures souhaite gérer ses locations et ses clients . On considère alors que les informations sur l’agence sont stockées dans le fichier **« agence.txt »** comme suit :



et on veut extraire ces infomations et les stocker dans un dictionnaire contenant les informations sur chaque location :

***Par exemple :*** *Location= {****100****: {nom\_client : "Ahmed",datelocation : "23/01/2022",immatriculation : "44A1233",Marque : "Kia"},****200****: {nom\_client : "Fatima",datelocation : "12/12/2021",immatriculation : "44A45466",Marque : "Ford"}}*

Les numéro de la location : les clés du dictionnaire.

Le nom du client , la date de la location , l’immatriculation de la voiture et sa marque sont représentés par un dictionnaire : Les valeurs du dictionnaire.

1. Ecrire un code qui permet d’ouvrir le fichier extraire tout les informations et les stocker dans un dictionnaire comme mentionné dans l’exemple **(4 pts).**
2. Ecrire une procédure **afficher\_location(Location)** qui reçoit un dictionnaire et qui permet d’afficher les informations de chaque location .**(3 pts)**
3. Ecrire une fonction **modifier\_location(Location, numéro)** qui reçoit un dictionnaire et un numéro et qui modifie les informations d’une location. **(3 pts)**
4. Ecrire une fonction **supprimer\_location(Location, numéro)** qui reçoit un dictionnaire et un numéro et qui permet de retirer la location dont son numéro est passé en paramètre.

Un message d’erreur sera affiché si le numéro est introuvable. **(3 pts)**

1. Ecrire une fonction **meilleur\_voiture(Location)** qui reçoit un dictionnaire et qui

Retourne l’immatriculation de la voiture la plus louée **(3 pts)**

1. Dans le programme principal, proposer à l’utilisateur un menu pour appeler les fonctions déclarées précédemment **(4 pts)**

**------------------ MENU ------------------------**

**1 – Afficher Locations**

**2 – Modifier Location**

**3 - Supprimer Location**

**4 – Afficher la voiture la plus louée**

**5 - Quitter**

**Entrez votre choix :**