

**EXAMEN DE FIN DE SEMESTRE1**

**Matière : INFORMATIQUE**

**Classes : 2<sup>ème</sup> Année BG      Durée : 2 h**

Barème : Exercice 1: 5 points, Exercice 2: 8 points, Exercice 3: 7points

**EXERCICE 1 :**

Soit le code Python suivant:

```
b=int(input("donner la valeur de b"))
m=int(input("donner la valeur de m"))
r = 1
while m > 0:
    if m % 2 == 1:
        r = r * b
    b = b * b
    m = m // 2
print(r)
```

1/ Quel est le résultat affiché :

A. Si b=2 et m=3

B. Si b=3 et m=3

2/ Quel est le rôle de cette fonction.

**EXERCICE 2 :**

L'hydrophobicité se définit comme un enchainement dans une protéine d'acides aminés dont les résidus sont hydrophobes. La succession de zones d'hydrophobicité dans une protéine définit le profil d'hydrophobicité de la molécule. Ce profil permet de repérer par exemple des régions potentiellement transmembranaires de la protéine étudiée. En effet, les régions transmembranaires, compte tenu de la nature des membranes biologiques, sont nécessairement hydrophobes.

1. Ecrivez une fonction en python **DegreHydro** qui prend en entrée une chaîne de caractères représentant une protéine et calcule son degré d'hydrophobicité, défini par la moyenne des indices d'hydrophobicité de ses acides aminés. On donne comme variable globale le dictionnaire des indices d'hydrophobicité pour chacun des acides aminés :

```
hydrophobicity = { "R": -4.5 , "Q": -3.5 , "Y": -1.3 , "G": -0.4 , "F": 2.8 , "K": -3.9 , "E":  
-3.5 , "W": -0.9 , "A": 1.8 , "L": 3.8 , "N": -3.5 , "H": -3.2 , "S": -0.8 , "M": 1.9 , "V":  
4.2 , "D": -3.5 , "P": -1.6 , "T": -0.7 , "C": 2.5 , "I": 4.5 } .
```

**Exemple :**

```
>>> proteine='ATGGC'
>>> hydroph = { "R": -4.5 , "Q": -3.5 , "Y": -1.3 , "G": -0.4 , "F": 2.8 , "K": -3.9 , "E":
-3.5 , "W": -0.9 , "A": 1.8 , "L": 3.8 , "N": -3.5 , "H": -3.2 , "S": -0.8 , "M": 1.9 , "V":
4.2 , "D": -3.5 , "P": -1.6 , "T": -0.7 , "C": 2.5 , "I": 4.5 } .
>>> print( DegreHydro ( proteine,n ) )
0.56
```

2. Ecrivez une fonction **Sous\_Sequences** qui prend en entrée une chaîne de caractère représentant la protéine, et qui renvoie la liste des sous-séquences obtenues par fenêtre glissante de taille donnée n.

Par exemple : pour la protéine de séquence 'KNDWII' et pour une fenêtre de 3, la fonction doit renvoyer la liste composée par les sous-séquences KND, NDW, DWI et WII.

```
>>> proteine='KNDWII'
>>> print(Sous_Sequences ( proteine,n))
['KND' , 'NDW' , 'DWI' , 'WII']
```

3. En utilisant les fonctions précédentes, écrivez une fonction **Profil\_Hydro** qui prend en entrée une chaîne de caractères représentant une protéine, et qui renvoie son profil d'hydrophobicité sous forme d'un dictionnaire : les degrés d'hydrophobicité de ses sous-séquences calculés par fenêtre glissante, représentent les valeurs du dictionnaire et les sous séquences représentent les clés.

**Exemple :** pour la protéine de séquence 'KNDWII' et pour une fenêtre de 3, la fonction doit calculer la moyenne des degrés d'hydrophobicité sur la sous-séquence KND, puis sur la sous-séquence NDW, puis sur DWI et enfin sur WII.

```
>>> proteine='KNDWII'
>>> n=3
>>> print( Profil_Hydro( proteine, n))
{'KND' :-3.63, 'NDW' :-2.63 , 'DWI' :0.033, 'WII' : 2.7}
```

**NB :** Vous pouvez utiliser les tranches (slices) comme l'exemple suivant :

```
>>> ch='examen'
>>> print(ch[0 :4])
exam
```

### EXERCICE 3 :

Soit le schéma relationnel de la base de données d'un hôpital défini par les relations suivantes :

- Patient (IdPatient, Nom, Prenom, Age, Adresse, Date\_Entree, Date\_Sortie)

La table *Patient* contient tous les patients actuellement présents à l'hôpital.

- **IdPatient** : identifiant du patient (entier), clé primaire.
- **Nom** : nom du patient (chaîne de caractères).
- **Prenom** : prénom du patient (chaîne de caractères).
- **Age** : âge du patient (entier).
- **Adresse** : adresse du patient (chaîne de caractères).
- **Date\_Entree** : date d'entrée du patient (chaîne de caractères),  
Exemple: "14-12-2017".
- **Date\_Sortie** : date de sortie du patient (chaîne de caractères égale NULL si la date de sortie est inconnue).

- Plat (IdPlat, Nom, Calories)

La table *Plat* contient tous les plats servis par la cuisine de l'hôpital.

- **IdPlat**: identifiant du plat (entier), clé primaire.
- **Nom** : nom du plat (chaîne de caractères).
- **Calories** : nombre total de calories (entier strictement positif).

- Servi (IdPlat, IdPatient, Date, Quand)

La table *Servi* énumère les plats identifiés par IdPlat servis, à une date donnée, aux patients identifiés par IdPatient.

- **IdPlat**: identifiant du plat (entier), clé primaire.
- **IdPatient**: identifiant du patient (entier), clé primaire.
- **Date** : date à laquelle le plat a été servi (chaîne de caractères).
- **Quand** : caractère indiquant quand le plat a été servi ; Quand = 'P' pour le petit déjeuner ; Quand = 'M' pour le repas de midi et Quand = 'S' pour le repas du soir.

- Interdit (IdPlat, IdPatient)

La table *Interdit* indique les plats, identifiés par IdPlat, interdits aux patients identifiés par IdPatient.

**Travail demandé :**

Ecrivez les requêtes SQL :

1. Déterminer tous les patients de l'hôpital.
2. Déterminer les identifiants, les noms, les prénoms et les dates d'entrée des patients âgés entre 25 et 50 ans.
3. Déterminer les identifiants, les noms, les prénoms et les dates d'entrée des patients dont la date de sortie est "14-12-2017".
4. Déterminer les identifiants des plats interdites pour le patient dont son identifiant est 546.
5. Déterminer les identifiants des plats dont le nom est "salade".
6. Déterminer les identifiants des patients qui ont été servis le "14-12-2017".