DS 2 - Informatique

Exercice 1 On considère l'équation différentielle :

$$\begin{cases} y' + \dfrac{1}{t+1}y = 1 \\ y(0) = 0 \end{cases}$$

1. Donner une approximation de $y(t+h)$ lorsque h est proche de 0 en fonction de h, t et $y(t)$.
2. En utilisant la méthode d'Euler, écrire une fonction de paramètres $t_{\max}$ et N qui donne la liste des valeurs de y pour t variant de 0 à $t_{\max}$ avec un pas de $h = \frac{t_{\max}}{N}$.

Exercice 2 (Chute d'un objet avec frottements quadratiques)
L'application du principe fondamental de la dynamique conduit à l'équation différentielle suivante :

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -Av^2 + g$$

L'objet possède une vitesse initiale nulle ($v(0) = 0$), $A = 2,03 \times 10^{-3}$SI et $g = 9,81$N.kg^{-1}.

1. Donner une approximation de $v(t+h)$ lorsque h est proche de 0 en fonction de h, t et $v(t)$.
2. Compléter le programme suivant permettant obtenir la vitesse de l'objet de 0 à $t_{\max}$ tous les dixièmes de seconde grâce à l'algorithme d'Euler.

```
A=2.03e-3
g=9.81
h=0.1

def f(v):
    return -A*v**2+g

def euler(tmax):
    t=0
    v=0
    V=[v]
    T=[t]
    while t<tmax:
        v= --------------------
        t= --------------------
          ---------------------
          ---------------------
    return T,V
```

3. Écrire les instructions permettant d'afficher la courbe de la vitesse en fonction du temps de 0 à 20 secondes (en utilisant la fonction ci-dessus).
4. On note $z(t)$ l'altitude de l'objet à l'instant t. On rappelle que la dérivée de z est $-v$.
   (a) Donner une approximation de $z(t+h)$ lorsque h est proche de 0 en fonction de h, t et $v(t)$.
   (b) Écrire une fonction `euler2` d'arguments z_0 (l'altitude initiale de l'objet) et $z_1 < z_0$, qui renvoie l'altitude de l'objet tous les dixièmes de seconde jusqu'à ce que l'objet atteigne l'altitude z_1.

Exercice 3 (Les pièces)
On dispose de pièces dont les valeurs en centimes sont données dans la liste (variable globale) :

```
Valeurs =[1,2,5,10,20,50,100,200]
```

1. On cherche à rendre la monnaie à un client en donnant le moins de pièces possibles. L'algorithme est le suivant : à chaque itération, on cherche la pièce de plus grande valeur possible.

   Compléter la fonction `glouton` de paramètre `S`, la somme à rendre au client (en centimes) et qui renvoie la liste des pièces données au client.

```
def glouton(S):
    Pieces=  ----------------------------
    S0=  ----------------------------
    k=  ----------------------------
    while S0<S:
        if    ----------------------------
            S0=  ----------------------------
            Pieces.append(Valeurs[k])
        else:
            ----------------------------
    return Pièces
```

2. Le client a un portemonnaie avec des pièces dont les valeurs sont parmi celles de la liste `Valeurs`.

   Le nombre de chaque pièce est donné dans une liste de longueur 8. Si la liste est `L`, `L[0]` est le nombre de pièces de 1 centime, `L[1]` est le nombre de pièces de 2 centimes, etc.

   (a) Écrire une fonction `portemonnaie` de paramètre `L` (représentant le portemonnaie), qui renvoie la valeur contenu dans le portemonnaie

   (b) Écrire une fonction `achat` de paramètres `S` (représentant une somme à payer en centimes) et `L` (représentant le portemonnaie) qui renvoie :
   — -1 s'il n'a pas assez d'argent,
   — 0 s'il n'a pas la somme exacte,
   — le nombre de pièces nécessaires pour payer la somme (sans rendu de monnaie).

Exercice 4
Considérons n intervalles $[a_i, b_i]$, $0 \leqslant i \leqslant n-1$ classés de sorte que $b_0 \leqslant b_1 \leqslant \ldots \leqslant b_{n-1}$.

Utiliser un algorithme glouton pour obtenir un ensemble maximal d'intervalles qui ne se chevauchent pas. L'ensemble des intervalles est stocké dans une liste de couples.

Par exemple :

```
L=[(1, 4),(3, 5),(0, 6),(5, 7),(3, 8),(5, 9),(6, 10),(8, 11),(8, 12),(2, 13),(12, 14)]
```

Dans ce cas, on obtient la liste :

```
M=[(1, 4), (5, 7), (8, 11), (12, 14)]
```

On peut penser au problème d'optimisation de l'occupation d'une salle (connaissant les heures de début et de fin d'activités pouvant avoir lieu)