DS d'informatique n°4 (durée : 2h)

Calculatrice autorisée. Quelques questions demandent d'écrire du code python, vous veillerez autant que possible à utiliser une syntaxe valide et à indenter correctement le code : utilisez des lignes verticales pour marquer les différents niveaux d'indentation.

Lorsqu'une question demander de calculer une complexité, la réponse devra être systématiquement justifiée.

Exercice 1 : Chiffrage d'un texte

En python, une chaîne de caractère est une liste de caractères. La majeur partie des opérations agissant sur une liste peuvent également agir sur une chaîne de caractères. Par exemple :

- `ch="chaine de test"` stocke dans la variable `ch` la chaîne de caractères `chaine de test`.
- `len(ch)` renvoie le nombre de caractères contenus dans la chaîne, ici `len(ch)=14`
- `ch[i]` permet d'accéder au caractère en position `i` dans la chaîne, par exemple `ch[2]` renvoie `'a'` et `ch[6]` renvoie `' '`. Par contre on ne peut pas modifier un caractère individuellement, par exemple `ch[2]='a'` renvoie une erreur.
- Pour extraire une sous-chaîne de la chaîne `ch` comprise entre les positions `i` (inclus) et `j` (exclus), on utilise `ch[i:j]`. Par exemple `ch[1:6]='haine'`
- Pour concaténer (mettre à la suite) deux chaînes de caractères, on utilise l'opérateur `+`. Par exemple `ch+"34"` renvoie `"chaine de test34"`.

1. Écrire une fonction `contient(ch,c)` qui prend en paramètre une chaîne de caractères `ch` et un caractère `c` et qui renvoie `True` si `ch` contient `c` et `False` sinon.
   Par exemple `contient("Bonjour",'o')` renvoie `True` alors que `contient("Bonjour",'z')` renvoie `False`.
2. Écrire une fonction `occurences(ch,c)` qui prend en paramètre une chaine de caractères `ch` et un caractère `c` et qui renvoie le nombre d'occurences du caractère `c` dans la chaine `ch`.
   Par exemple `occurences("Bonjour",'o')` renvoie 2.
3. Donner la complexité de la fonction `occurences(ch,c)` en fonction de n =`len(ch)`.
4. On supposera connue la fonction `occurences(ch,c)` définie à la question précédente. On cherche à déterminer le nombre d'occurences de chacun des caractères d'une liste `cars` dans une chaîne de caractère `ch`. Écrire une fonction `listeOccurences(ch,cars)` qui renvoie une liste contenant le nombre d'occurences de chacun des caractères de la liste `cars` dans la chaîne `ch`.
   Par exemple `listeOccurences("Bonjour",['a','o','u'])` renvoie la liste `[0,2,1]`.
5. Donner la complexité de la fonction `listeOccurences(ch,cars)` en fonction de n =`len(ch)` et de m =`len(cars)`.
6. L'instruction `list(ch)` renvoie une liste contenant les caractères de la chaîne de caractères `ch`. Pour une chaîne de caractères `ch`, l'instruction `ch.lower()` renvoie la chaîne `ch` dont tous les caractères ont été convertis en minuscule. On définit la fonction suivante :

```
def truc(chaine):
  lettres = "abcdefghijklmnopqrstuvwxyz"
  liste = list(lettres)
  ch=chaine.lower()
  occ = listeOccurences(ch,liste)
  index = 0
  for i in range(len(liste)):
    if(occ[i]>occ[index]):
      index = i
  return liste[index]
```

   Expliquer l'utilité de la fonction `truc` suivante et donner la valeur renvoyée par `truc("Bonjour tout le monde")`
7. Écrire une fonction `position(ch,c)` qui renvoie la position de la première apparition du caractère `c` dans la chaîne `ch`. Cette fonction renvoie -1 si le caractère `c` n'est pas contenu dans la chaîne `ch`.
8. On souhaite crypter un texte pour le rendre illisible par une personne ne possédant pas la clé de décryptage. Pour cela on utilise la fonction suivante :

```
def crypt(chaine,cle):
  lettres = "abcdefghijklmnopqrstuvwxyz"
  ch = chaine.lower()
  res = ""
  for c in ch:
    index = position(lettres,c)
    if index>=0:
      index = index+cle
      if index>len(lettres):
```

```
        index = index-len(lettres)
      res = res+lettres[index]
    else:
      res = res+c
  return res
```

La fonction `position(ch,c)` est celle définie à la question précédente. Quel est le type des paramètres `chaine` et `cle` ? Expliquer en quelques mots le fonctionnement de la fonction `crypt` et donner la valeur renvoyée par

```
crypt("message secret", 8)
```

9. Écrire une fonction `decrypt(chaine,cle)` qui effectue l'opération inverse, c'est à dire qui étant donnée un message crypté contenu dans `chaine` et une clé contenue dans `cle` retourne le message décrypté.
10. Sachant que la lettre 'e' est la lettre la plus fréquente de la langue française, quel algorithme pourrait-on utiliser pour trouver la clé de cryptage d'un message codé (On ne demande pas les détails du code python utilisé, uniquement le principe de la méthode). On pourra utiliser les différentes fonctions définies aux questions précédentes.
    Discuter de la sécurité de ce système de cryptage. Quelle amélioration proposeriez vous ?

Exercice 2 : Méthode d'Euler

On donne ci-dessous une fonction permettant de résoudre une équation différentielle de la forme $y' = f(y,t)$ par la méthode d'Euler :

```
def fct(y,t):
    return -4*y+5

def euler(f,t0,y0,tmax,h):
    valeurs=[y0]
    temps=[t0]
    y=y0
    t=t0
    while(t<tmax):
        y=y+h*f(y,t)
        t=t+h
        valeurs.append(y)
        temps.append(t)
    return temps,valeurs
```

On résout l'équation différentielle en executant l'instruction `tps,vals = euler(fct,0,10,2,0.01)`.

1. Quelle est l'équation différentielle que l'on résout de cette manière ?
2. À quoi correspondent `t0`, `y0`, `tmax` et `h` ? Comment doit-on choisir `h` pour que la solution trouvée soit précise ? Quel problème cela peut-il poser ?
3. Donner les 5 premières valeurs contenues dans les listes `tps` et `vals` après l'exécution de la commande ci-dessus.
4. Écrire la fonction python à utiliser pour résoudre l'équation différentielle : $y' - 2y^2 + 3y = 2$.
5. Indiquer comment il faudrait modifier le programme précédent pour résoudre une équation différentielle d'ordre 2 du type $y'' = f(y', y, t)$. On ne demande pas d'écrire le programme correspondant mais d'en expliquer le principe général.

Exercice 3 : Intégration numérique

Écrire une fonction `integRectangles(f,a,b,n)` qui calcule numériquement la valeur de l'intégrale de la fonction f (dont les valeurs sont calculées par une fonction `f(x)`) entre `a` et `b` en utilisant `n` rectangles. Vous ferez un schéma permettant d'illustrer la méthode sur lequel vous ferez apparaître les grandeurs importantes qui interviennent dans votre algorithme.