

Master 1 Humanités numériques – Algorithmique et programmation

Examen sur machine – 14/01/2019

J. Darmont – <http://eric.univ-lyon2.fr/jdarmont/>

Durée : 2 heures – Documents autorisés – Barème fourni à titre indicatif

Rendu : Ensemble de fichiers rassemblés dans un répertoire à votre nom

Exercice 1 (2 points)

Dans un fichier texte, définir un algorithme en quelques mots.

Exercice 2 (2 points)

Écrire dans un fichier texte l'algorithme de la fonction *pc80*, qui prend en paramètres deux nombres réels *r1* et *r2* et effectue les opérations suivantes :

1. calculer le pourcentage *p* défini par la division de *r1* par *r2*, le tout multiplié par 100 ;
2. si *p* est supérieur ou égal à 80, la fonction *pc80* doit retourner la valeur booléenne Vrai ;
3. sinon, la fonction *pc80* doit retourner la valeur booléenne Faux.

Exercice 3 (3 points)

Écrire un programme Python permettant de :

1. saisir au clavier une chaîne de caractères *ch* ;
2. stocker dans une variable *demiLongueur* la longueur de *ch* divisée par 2 convertie en nombre entier ;
3. stocker dans deux variables *ch1* et *ch2* les sous-chaînes correspondant au début de *ch* jusqu'à *demiLongueur* et à la fin de *ch* à partir de *demiLongueur*, respectivement (si vous n'avez pas répondu à la question 2, fixez *demiLongueur* = 3) ;
4. afficher *ch1* et *ch2*.

Dans les exercices suivants, si vous ne parvenez pas à écrire les fonctions, faites un programme classique à la place.

Exercice 4 (3 points)

1. Écrire une fonction Python nommé *totLen* prenant en paramètre une liste de chaînes de caractères et retournant la longueur totale de toutes les chaînes.
2. Écrire un programme principal qui crée une liste de chaînes de caractères, appelle la fonction *totLen* et affiche la longueur totale des chaînes de la liste.

Exercice 5 (4 points)

1. Écrire une fonction Python nommée *topK* prenant en paramètres une liste quelconque et un entier *k*. *topK* doit afficher les *k* plus grandes valeurs de la liste. Pour cela, il faut trier la liste par ordre décroissant en passant le paramètre *reverse = True* à la méthode *sort()*. Si la taille *t* de la liste est inférieure à *k*, fixer *k* à *t* pour éviter toute erreur.
2. Écrire un programme principal qui crée une liste de nombres et une liste de chaînes de caractères, puis qui appelle la fonction *topK* pour chaque liste avec deux valeurs de *k* différentes, dont une plus petite que la taille de la liste correspondante.

Exercice 6 (6 points)

1. Écrire une fonction Python nommée *formater* prenant en paramètres les noms d'un fichier d'entrée (*ficEntree*) et d'un fichier de sortie (*ficSortie*). L'objectif est de reformater *ficEntree* en transformant toutes les lignes se terminant par un point d'interrogation en majuscules et en ajoutant un saut de ligne après chaque ligne. Le résultat soit être sauvegardé dans *ficSortie*. Pour cela :
 - a. charger le contenu de *ficEntree* dans une liste ;
 - b. parcourir la liste des lignes du texte :
 - i. si une ligne se termine par « ? », la transformer en majuscules. Attention : le dernier caractère d'une ligne est en fait `\n`, qui est invisible. Les « ? » éventuels sont donc en avant-dernière position dans la ligne ;
 - ii. ajouter un saut de ligne à la ligne courante ;
 - iii. ajouter la ligne modifiée dans une liste résultat que vous aurez préalablement initialisée à vide ;
 - c. écrire le contenu de la liste résultat dans *ficSortie*.
2. Écrire un programme principal qui appelle la fonction *formater* en lui passant en paramètres le nom du fichier texte *rominou2.txt* (créé avec le générateur de texte français aléatoire ROMINO¹ et disponible à l'adresse <http://eric.univ-lyon2.fr/jdarmont/docs/rominou2.txt>) et le nom d'un fichier résultat à votre convenance (par exemple, *rominou2-formate.txt*).

¹ <https://github.com/sparusaurata/rominou>

Correction Exercice 1

Un algorithme est une suite d'instructions ordonnées qui permet, à partir de données en entrée, de résoudre un problème/un calcul et d'en fournir le résultat en sortie.

Correction Exercice 2

Fonction pc80(r1, r2 : Réels) : Booléen

Var p : Réel

Début

{Calcul du pourcentage}

$p \leftarrow 100 \times r1 \div r2$

{Test du seuil de 80 %}

Si $p \geq 80$ alors

Retourner Vrai

Sinon

Retourner Faux

Fin si

Fin

Correction Exercice 3

Saisie de la chaîne

ch = input("Saisir une chaîne : ")

Découpage de la chaîne

demiLongueur = int(len(ch)/2)

ch1 = ch[:demiLongueur]

ch2 = ch[demiLongueur:]

Affichage des sous-chaînes

print("1re moitié :", ch1)

print("2e moitié :", ch2)

Correction Exercice 4

def totLen(liste):

Initialisation de la longueur totale

total = 0

Parcours de la liste de chaînes

for ch in liste:

Ajout de la longueur de la chaîne courante au total

total += len(ch)

return total

Programme principal

```
listeChaines = ["TF1", "France 2", "France 3", "Canal+", "Arte", "M6"]
print("Longueur totale :", totLen(listeChaines))
```

Correction Exercice 5

```
def topK(liste, k):
    # Test sur la longueur de la liste
    t = len(liste)
    if k > t:
        k = t
    # Tri de la liste par ordre décroissant
    liste.sort(reverse = True)
    # Parcours de la liste k fois
    for i in range(0, k):
        print(liste[i])
```

Programme principal

```
listeN = [3.14, 42, -8, 15, 10, 20, 33]
listeC = ["Adava", "Kevadra", "Zorro", "Renard", "Speedy", "Gonzales"]
topK(listeN, 3)
topK(listeC, 20)
```

Correction Exercice 6

```
def formater(ficEntree, ficSortie):
    # Chargement du fichier dans une liste
    ficTexte = open(ficEntree, "r")
    listeTexte = ficTexte.readlines()
    ficTexte.close()
    # Traitement de listeTexte et création de la liste résultat
    listeRes = [ ]
    for ligne in listeTexte:
        # Mise en majuscules des questions
        if ligne[len(ligne) - 2] == "?":
            ligne = ligne.upper()
        # Ajout d'un saut de ligne
        ligne += "\n"
        # Ajout au résultat
        listeRes.append(ligne)
    # Ecriture du fichier formaté
    ficRes = open(ficSortie, "w")
    ficRes.writelines(listeRes)
    ficRes.close()
```

Programme principal

```
formater("rominou2.txt", "rominou2-formate.txt")
```