



Master 1 Humanités numériques – Algorithmique et programmation

Examen sur machine – 15/01/2018

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)

Décrire (dans un fichier texte) en quelques mots le processus de conception utilisé au cours de cet enseignement pour produire programme informatique.

### Exercice 2 (2 points)

Écrire un programme Python permettant de :

- saisir au clavier successivement deux chaînes de caractères ;
- stocker dans une variable la concaténation de la première chaîne transformée en minuscules et de la seconde transformée en majuscules ;
- afficher la variable résultat.

### Exercice 3 (3 points)

Écrire (dans un fichier texte) l'algorithme d'une procédure prenant en paramètres deux nombres réels  $r1$  et  $r2$  et effectuant les opérations suivantes :

1. calculer la soustraction  $r$  de  $r1$  par  $r2$  ;
2. si  $r$  est négative, alors afficher simplement la valeur de  $r$  (le signe « - » apparaîtra automatiquement) ;
3. sinon, afficher un « + » devant la valeur de  $r$ .

### Exercice 4 (3 points)

Soit une liste de notes sur 20.

1. Écrire un programme Python permettant de calculer la moyenne des notes (somme des notes ÷ nombre de notes).

2. Transformer le programme en une fonction qui prenant en paramètre une liste de nombres et en retournant la moyenne. Donner un exemple de programme principal qui appelle cette fonction et affiche la moyenne.

### Exercice 5 (3 points)

Soit une liste de chaînes de caractères quelconques. Écrire un programme Python qui vérifie la longueur de chacune des chaînes de la liste. Si l'une d'entre elle dépasse 8 caractères, elle doit être tronquée pour ne comprendre que 8 caractères exactement (les 8 premiers). En fin de programme, afficher la liste pour vérification.

### Exercice 6 (7 points)

On dispose de trois fichiers textes :

- *rominou.txt* contient un texte à traiter, créé avec le générateur de texte français aléatoire ROMINO<sup>1</sup> ;
- *ponctuation.txt* contient une liste de signes de ponctuation ;
- *mots-outils.txt* contient une liste de mots outils (qui sont en général retirés des textes lors d'analyses).

L'objectif de cet exercice est d'écrire un programme permettant de retirer du texte contenu dans le fichier *rominou.txt* tous les signes de ponctuation et les mots outils contenus dans les fichiers *ponctuation.txt* et *mots-outils.txt*, respectivement. Le résultat devra être sauvegardé dans le nouveau fichier *rominou-traite.txt*.

Téléchargement des fichiers :

- <http://eric.univ-lyon2.fr/~jdarmont/docs/rominou.txt>
- <http://eric.univ-lyon2.fr/~jdarmont/docs/ponctuation.txt>
- <http://eric.univ-lyon2.fr/~jdarmont/docs/mots-outils.txt>

Indications :

1. Charger le contenu des trois fichiers dans des listes.
2. Parcourir la liste les lignes du texte :
  - a. remplacer successivement dans la ligne courante le contenu de la liste des signes de ponctuation par des espaces ;
  - b. remplacer successivement dans la ligne courante le contenu de la liste des mots outils par des espaces (afin de ne pas enlever de portions de mots dans le texte, ajouter une espace à gauche et à droite des mots outils lors du remplacement) ;
  - c. ATTENTION : les signes de ponctuation et les mots outils contiennent le caractère `\n`, qu'il convient de supprimer avant d'effectuer les remplacements en a et b ;
  - d. ajouter la ligne modifiée dans une liste résultat.
3. Écrire le contenu de la liste résultat dans le fichier *rominou-traite.txt*.

---

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

## Correction Exercice 1

L'analyse d'un problème est la première étape dans la production d'un programme qui le résout. Elle mène à concevoir un algorithme, qui peut être exprimé dans un langage algorithmique textuel ou graphique (Blockly), qui est ensuite traduit dans un langage de programmation (Python). C'est ce dernier qui permet l'exécution effective du programme.

## Correction Exercice 2

**# Saisie des chaînes**

```
c1 = input("chaîne 1 : ")
```

```
c2 = input("chaîne 2 : ")
```

**# Chaîne résultat**

```
c3 = c1.lower() + c2.upper()
```

**# Affichage**

```
print("résultat :", c3)
```

## Correction Exercice 3

Procédure NégatifPositif( $r_1, r_2$  : Réels)

Var  $r$  : Réel

Début

{Calcul de la différence}

$r \leftarrow r_1 - r_2$

Si  $r < 0$  alors { $r$  est négatif, rien à faire}

Écrire( $r$ )

Sinon { $r$  est positif, on ajoute un + devant  $r$ }

Écrire("+",  $r$ )

Fin si

Fin

## Correction Exercice 4

```
def moyenne(listeNombres):
```

```
    somme = 0
```

```
    for n in listeNombres:
```

```
        somme += n
```

```
    return somme / len(listeNombres)
```

**# Programme principal**

```
listeNotes = [4, 13.5, 12.5, 16, 9, 14.5]
```

```
print("moyenne :", moyenne(listeNotes))
```

## Correction Exercice 5

### # Initialisation de la liste

```
liste = ["ALPENSTOCK", "AMOK", "ANORAK", "ANTIQUARK", "APPARATCHIK", "ARACK", "ARAK", "ARAWAK", "ATTIKAMEK"]
```

### # Parcours de la liste

```
for i in range(0, len(liste)):
    if len(liste[i]) > 8:          # Si la chaîne a plus de 8 caractères
        liste[i] = liste[i][0:8]  # On la remplace par ses 8 premiers
```

### # Affichage

```
print(liste)
```

## Correction Exercice 6

### # Chargement des fichiers dans les listes

```
ficTexte = open("rominou.txt", "r")
listeTexte = ficTexte.readlines()
ficTexte.close()
```

```
ficMots = open("mots-outils.txt", "r")
listeMots = ficMots.readlines()
ficMots.close()
```

```
ficPonct = open("ponctuation.txt", "r")
listePonct = ficPonct.readlines()
ficPonct.close()
```

### # Traitement de listeTexte et création de la liste résultat

```
listeRes = [ ]
```

```
for ligne in listeTexte:
    # Suppression des signes de ponctuation
    for ponct in listePonct:
        ligne = ligne.replace(ponct.replace("\n", ""), " ")
    # Suppression des mots outils
    for mot in listeMots:
        ligne = ligne.replace(" " + mot.replace("\n", "") + " ", " ")
    # Ajout au résultat
    listeRes.append(ligne)
```

### # Ecriture du fichier résultat

```
ficRes = open("rominou-traite.txt", "w")
ficRes.writelines(listeRes)
ficRes.close()
```