

Master 1 Humanités numériques – Algorithmique et programmation Examen sur machine – 09/01/2023

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

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

Rendu : Ensemble de fichiers rassemblés dans un répertoire compressé

Exercice 1 (2 points)

Dans un fichier texte, décrire en quelques mots le processus de conception utilisé au cours de cet enseignement pour produire programme informatique.

Exercice 2 (4 points)

Dans un fichier texte, écrire l’algorithme Prime qui effectue les opérations suivantes :

- 1 saisir au clavier deux variables *salaire* et *performance* ;
- 2 si la *performance* est « A », multiplier le *salaire* par 1,5 ;
- 3 sinon, si la *performance* est « B », multiplier le *salaire* par 1,2 ;
- 4 sinon, multiplier le *salaire* par 1,05 ;
- 5 afficher à l’écran la valeur modifiée du *salaire*.

Exercice 3 (4 points)

On souhaite dessiner un éventail multicolore à l’aide d’un programme Python.

- 1 Importer toutes les fonctions du module turtle.
- 2 Créer une liste *couleurs* contenant les couleurs suivantes : black, blue, cyan, aquamarine, turquoise, green, yellow, orange, tomato, red, brown, chocolate, tan, wheat, pink, violet, orchid, magenta, purple.
- 3 Paramétrer la tortue :
 - 3.a cacher la tortue
 - 3.b passer la vitesse du tracé à « fastest »
 - 3.c fixer la largeur du trait à 50 pixels.
- 4 Pour chaque couleur de la liste *couleurs* :
 - 4.a abaisser le stylo et faire avancer la tortue de 300 pixels
 - 4.b relever le stylo et faire reculer la tortue de 300 pixels
 - 4.c tourner à gauche avec un angle de 10 degrés.

Fonctions utiles du module turtle :

- `backward(d)`, où d est la distance parcourue en arrière (en pixels)
- `forward(d)`, où d est la distance parcourue en avant (en pixels)
- `hideturtle()` cache la tortue
- `left(a)`, où a est l'angle en degrés
- `pencolor(c)`, où c est la couleur du stylo
- `pendown()` abaisse le stylo
- `penup()` relève le stylo
- `speed(s)`, où s est la vitesse du tracé
- `width(w)`, où w est la largeur du trait

Exercice 4 (4 points)

Écrire une fonction Python nommée *truncList* qui prend en paramètres formels une *liste* de chaînes de caractères et une *taille* (nombre entier). *truncList* doit vérifier la longueur de chacune des chaînes de la liste. Si une chaîne dépasse *taille* caractères, elle doit être tronquée pour ne comprendre que *taille* caractères exactement (les *taille* premiers). En fin de fonction, afficher la *liste* pour vérification.

Écrire un programme principal :

- créer une liste de chaînes de caractères quelconques nommée *maListe*
- appeler la fonction *truncList* avec comme paramètres réels *maListe* et une taille au choix (8, par exemple).

Exercice 5 (6 points)

On dispose de deux fichiers textes : *python.txt*, qui est un texte à pré-traiter¹, et *ponctuation.txt*, qui contient une liste de signes de ponctuation (étonnant, non ?). L'objectif de cet exercice est d'écrire un programme permettant de retirer du texte contenu dans le fichier *python.txt* tous les signes de ponctuation contenus dans le fichier *ponctuation.txt*. Le résultat doit être sauvegardé dans le nouveau fichier *pythonTraite.txt*.

Téléchargement des fichiers :

- <https://eric.univ-lyon2.fr/jdarmont/docs/python.txt>
- <https://eric.univ-lyon2.fr/jdarmont/docs/ponctuation.txt>

Indications :

1. Charger le contenu des deux fichiers dans des listes.
2. Parcourir la liste des lignes du texte :
 - a. remplacer successivement dans la ligne courante le contenu de la liste des signes de ponctuation par des espaces ;
 - b. ATTENTION : les signes de ponctuation contiennent le caractère « `\n` », qu'il convient de supprimer avant d'effectuer les remplacements en 2a ;
 - c. ajouter la ligne modifiée dans une liste résultat.
3. Écrire le contenu de la liste résultat dans le fichier *pythonTraite.txt*.

¹ Issu de [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage))

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, exprimé dans un langage algorithmique textuel, qui est ensuite traduit dans un langage de programmation (Python). C'est ce dernier qui permet l'exécution effective du programme.

Correction Exercice 2

Algorithme Prime

```
Var salaire : Réel
Var performance : Chaîne
```

Début

```
{Saisies au clavier}
Lire("Salaire :", salaire)
Lire("Performance :", performance)
{Modifications du salaire en fonction de la performance}
Si performance = "A" alors
    salaire ← salaire × 1,5
Sinon si performance = "B" alors
    salaire ← salaire × 1,2
Sinon
    salaire ← salaire × 1,05
Fin si
{Affichage du salaire modifié}
Écrire("Salaire final :", salaire)
```

Fin

Correction Exercice 3

Importation du module turtle

```
from turtle import *
```

Liste des couleurs

```
couleurs = ["black", "blue", "cyan", "aquamarine", "turquoise", "green", "yellow",
"orange", "tomato", "red", "brown", "chocolate", "tan", "wheat", "pink", "violet",
"orchid", "magenta", "purple"]
```

Paramètres de la tortue

```
hideturtle()
speed("fastest")
width(50)
```

Eventail

```
for couleur in couleurs:
    pencolor(couleur)
    pendown()
    forward(300)
```

```
penup()
backward(300)
left(10)
```

Correction Exercice 4

```
# Fonction truncList
def truncList(liste, taille):
    # Parcours de la liste
    for i in range(0, len(liste)):
        if len(liste[i]) > taille:           # Si la chaîne a plus de taille caractères
            liste[i] = liste[i][:taille]    # On la remplace par ses taille premiers
    # Affichage
    print(liste)

# Programme principal
# Initialisation de la liste
maListe = ["ALPENSTOCK", "AMOK", "ANORAK", "ANTIQUARK", "APPARATCHIK",
"ARACK", "ARAK", "ARAWAK", "ATTIKAMEK"]
# Appel à truncList
truncList(maListe, 8)
```

Correction Exercice 5

```
# Chargement des fichiers dans les listes
ficTexte = open("python.txt", "r")
listeTexte = ficTexte.readlines()
ficTexte.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", ""), " ")
    # Ajout au résultat
    listeRes.append(ligne)

# Ecriture du fichier résultat
ficRes = open("pythonTraite.txt", "w")
ficRes.writelines(listeRes)
ficRes.close()
```