

Master 1 Humanités numériques – Algorithmique et programmation

Examen sur machine – 06/01/2020

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 compressé à votre nom

Exercice 1 (3 points)

Dans un fichier texte, expliquer en quelques mots la différence que vous faites entre le langage algorithmique textuel, Blockly et Python.

Exercice 2 (3 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 1,4 ;
3. sinon, si la *performance* est « B », multiplier le *salaire* par 1,2 ;
4. sinon, multiplier le *salaire* par 1,1 ;
5. afficher à l'écran la valeur modifiée du *salaire*.

Exercice 3 (4 points)

I. Écrire un programme Python permettant de :

1. créer une liste *listeCh* de chaînes de caractères quelconques et de tailles diverses ;
2. afficher à l'écran chaque élément *ch* de la liste ;
3. toutefois, si la taille de la chaîne *ch* dépasse 4 caractères, n'afficher que les 4 premiers caractères de *ch*.

II. Transformer l'affichage de la liste en une fonction nommée *affListe* prenant en paramètre un nombre *k*.

1. Remplacer la limite de 4 caractères par *k* caractères.
2. Écrire un programme principal qui appelle la fonction *affListe* pour quelques valeurs de *k*.

Exercice 4 (4 points)

Soit le règlement de scolarité suivant : un-e étudiant-e réussit son année s'il/elle obtient plus de 10/20 aux deux semestres S1 et S2. Écrire un programme Python permettant de :

1. saisir au clavier successivement les notes *noteS1* et *noteS2* ;
2. si les deux notes sont supérieures ou égales à 10, afficher le message « Année réussie » ;
3. sinon, si *noteS1* (respectivement, *noteS2*) est inférieure à 10, afficher le message « Echec au S1 » (respectivement, « Echec au S2 »).

NB : si l'étudiant-e échoue aux deux semestres, les deux messages d'échec au S1 et au S2 doivent apparaître.

Exercice 5 (6 points)

I. Écrire une fonction Python nommée *supprPonctuation*, prenant en paramètre un nom de fichier texte (chaîne de caractères) nommé *fichier*. La fonction a pour objectif de supprimer tout signe de ponctuation du texte initial et de sauvegarder le résultat dans un fichier nommé *res-fichier*.

1. Créer une liste *ponctuation* constituée des chaînes de caractères suivantes : `"," "?" ";" "." ":" "!" "\n" "\t" "(" ")" "-" ""`.
2. Charger le contenu de *fichier* dans une liste nommée *lignes*.
3. Initialiser une liste de lignes modifiées nommée *res* à vide.
4. Pour chaque ligne de *lignes* :
 - o pour chaque signe de *ponctuation*, remplacer dans la ligne courante le signe de ponctuation par une chaîne vide ;
 - o ajouter la ligne modifiée à *res*.
5. Sauvegarder le contenu de *res* dans un fichier texte de nom *res-fichier*.

II. Écrire un programme principal qui appelle la fonction *supprPonctuation* en lui passant en paramètre le nom du fichier "rominou2.txt" (créé avec le générateur de texte français aléatoire ROMINO²), que vous pouvez télécharger à l'adresse <http://eric.univ-lyon2.fr/jdarmont/docs/rominou.txt>. Le fichier résultat devrait s'appeler "res-rominou2.txt".

¹ La notation `\` permet de placer le caractère `"` dans une chaîne délimitée par des caractères `"`. `\` est un caractère dit d'échappement.

² <https://github.com/sparusaurata/rominou>

Correction Exercice 1

Le langage algorithmique textuel permet d'analyser un problème et de l'écrire « sur papier » en amont de son implémentation informatique.

Blockly peut être considéré comme un langage algorithmique graphique. Le résultat peut être exécuté sur ordinateur, mais pas sauvegardé. Blockly permet toutefois de générer des programmes dans différents langages de programmation, dont Python.

Python est un langage de programmation à part entière, qui permet de déployer des applications sur différents systèmes informatiques.

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 × 1,4

Sinon si performance = "B" alors

 salaire ← salaire × 1,2

Sinon

 salaire ← salaire × 1,1

Fin si

{Affichage du salaire modifié}

Écrire("Salaire final :", salaire)

Fin

Correction Exercice 3

Partie I

```
listeCh = ["TF1", "France 2", "France 3", "Canal+", "Arte", "M6"]
```

```
for ch in listeCh:
    # Parcours de la liste
    if len(ch) > 4:
        # Cas où la longueur excède 4 caractères
        print(ch[0:5])
    # Affichage des 4 premiers caractères
    else:
        print(ch)
```

Partie II

Fonction

```
def affListe(k):
    listeCh = ["TF1", "France 2", "France 3", "Canal+", "Arte", "M6"]
    for ch in listeCh:
        if len(ch) > k:
            print(ch[0:k])
        else:
            print(ch)
```

Programme principal

```
affListe(2)
affListe(4)
affListe(8)
```

Correction Exercice 4

Saisie des notes

```
noteS1 = float(input("Note S1 : "))
noteS2 = float(input("Note S2 : "))
```

Test réussite

```
if noteS1 >= 10 and noteS2 >= 10:
    print("Année réussie")
else:
    # Test échec S1
    if noteS1 < 10:
        print("Echec au S1")
    # Test échec S2
    if noteS2 < 10:
        print("Echec au S2")
```

Correction Exercice 5

l-Fonction

```
def supprPonctuation(fichier):
    # Liste des signes de ponctuation
    ponctuation = [",", "?", ";", ".", ":", "!", "\\", "(", ")", "-", ""]
    # Lecture du fichier source
    f = open(fichier, "r")
    lignes = f.readlines()
    f.close()
    # Initialisation du résultat
    res = []
```

```
# Suppression de la ponctuation dans chaque ligne
for l in lignes:
    for p in ponctuation:
        l = l.replace(p, "")
    res.append(l)
# Ecriture du fichier résultat
f2 = open("res-" + fichier, "w")
f2.writelines(res)
f2.close()

# II-Programme principal
supprPonctuation("rominou2.txt")
```