

Session 2022

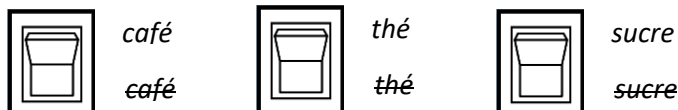
## Examen OC INFORMATIQUE

### Informations et consignes :

- Temps à disposition : 3 heures.
- Aucun document n'est autorisé.
- Calculatrice non programmable autorisée.
- Le nombre de points est indiqué pour chaque problème. Il y a 100 points au total.
- Utilisez un feuillet par problème.
- Vous devez écrire proprement au stylo ou à l'encre. La présentation est prise en compte pour la note.
- Écrivez votre nom sur chaque feuille.
- Rendez tous vos documents.

### Exercice 1 : Logique (15 points)

- a) Démontrer que :  $AB + A\bar{B}CD = AB + ACD$ .
- b) Dans une école, un distributeur de boissons chaudes fournit du café ou du thé, avec ou sans sucre. Toutes les boissons coûtent le même prix. Trois commutateurs permettent de sélectionner sa boisson : *café*, *thé* et *sucre*.



Une fois les commutateurs réglés, il faut introduire l'argent nécessaire dans l'appareil pour obtenir sa boisson. Il n'est pas possible de commander uniquement du sucre.

Si les commutateurs sont disposés d'une manière non valide, l'argent est rendu à l'utilisateur. Ainsi, si *café* et *thé* sont sélectionnés en même temps, l'argent est rendu. Il en va de même si rien n'est sélectionné.

Calculer et simplifier les fonctions de restitution de l'argent ( $R$ ) et de distribution du café ( $F$ ).

### Exercice 2 : Divers (25 points)

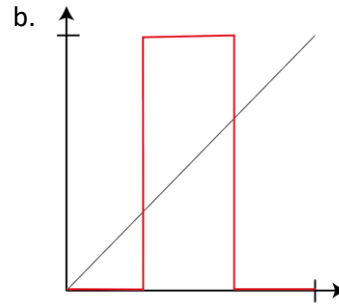
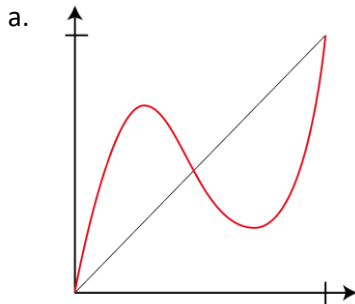
- a) Écrire un code en Python qui crée un dictionnaire faisant correspondre à chaque élément d'une liste son nombre d'apparitions.
- b) Effectuer un tri sur la liste  $li1 = [15, 4, 20, 17, 11, 8, 11, 16, 7, 14, 2]$ . Pour ce faire, il faut utiliser l'algorithme *quicksort*.
- c) Insérer les éléments de la liste  $li2 = [15, 4, 20, 17, 11, 8, 16, 7]$  dans un tas (min heap).
- d) Soit la liste  $li = [15, 4, 20, 17, 11, 8, 11, 16, 7, 14, 2, 7, 5, 17, 18, 4, 5, 13, 8]$ .  
On cherche à sélectionner 5 nombres de cette liste tels que la somme de ces nombres soit la plus grande possible. Contrainte supplémentaire : on ne peut pas prendre deux nombres voisins dans la liste.
- a. Donner une solution à ce problème en utilisant un algorithme glouton (approche heuristique).
  - b. Quelle autre méthode pourrait être utilisée ici ?

### Exercice 3 : Programmation et langages (15 points)

- Soit un nombre écrit en binaire. Écrire la table d'action de la machine de Turing qui ajoute un bit de parité (1 si le nombre de 1 est impair, 0 sinon).
- Citer trois paradigmes en programmation.
- Qu'est-ce que l'héritage ? A quel paradigme appartient-il ?

### Exercice 4 : Traitement d'image (20 points)

- Qu'est-ce qu'une courbe tonale ?
- Quelle différence y a-t-il entre « définition » et « résolution » ?
- Soit une image quelconque en couleur. Dessiner les courbes tonales permettant d'obtenir
  - le négatif de l'image ;
  - une augmentation de la luminosité ;
  - des nuances de rouge ;
- Soit une image quelconque en niveaux de gris. Qu'est-ce que les courbes tonales suivantes donneront comme résultat ?



### Exercice 5 : Python 3 (25 points)

Le *multant* d'un entier naturel  $n$  est un nombre entre 0 et 9. Par exemple, le multant de 77 est 8 car

$$77 \rightarrow 7 \times 7 = 49, 49 \rightarrow 4 \times 9 = 36, 36 \rightarrow 3 \times 6 = 18, 1 \times 8 = 8.$$

Le *multant* de  $n$  se calcule de la manière suivante : les chiffres du nombre  $n$  sont itérativement multipliés entre eux jusqu'à obtenir un seul chiffre.

- Écrire une fonction *multi* retournant la multiplication des chiffres d'un entier naturel donné.
- Écrire la fonction *multant1*, calculant le multant d'un entier naturel, de manière récursive.
- Écrire la fonction *multant2*, calculant le multant d'un entier naturel, de manière itérative.

À chaque entier naturel correspond donc un multant. Deux nombres différents peuvent avoir le même multant.

- Créer une liste permettant de classer les nombres de 0 à 10'000 suivant leur multant. Ainsi `mult[0]` donnera la liste des nombres dont le multant est 0.