Licence 3 - Culture Humaniste et Scientifique **Informatique**

Tout document autorisé

TP noté 2020–2021

Vous pouvez rendre votre travail de deux façons :

— Soit répondant aux questions **sur feuille**, puis en les **scannant** ou en les **photographiant**.

— Soit en répondant **directement dans le notebook**. Dans ce cas, pour les exercices 1, 2 et 3 qui demandent uniquement des réponses rédigées en français et pas de programme en Python, répondez directement dans les blocs de texte prévus à cet effet. Pour l'exercice 4, répondez dans le dernier bloc de code Python. Dans les deux cas, double-cliquez dans les blocs de réponse pour les éditer.

Même si vous rendez votre travail sur feuille, il est conseillé d'utiliser le notebook pour tester le code du dernier exercice.

À **15h30**, envoyez votre travail (copie scannée ou notebook) par mail aux adresses d'O. Baudon (olivier.baudon@labri.fr) et M. Zeitoun (mz@labri.fr) avec comme sujet : `[CHS] Rendu TP noté`.

Exercice 1

1. Quelle est la valeur représentée en base 2 par 1100 ? Détaillez le calcul vous ayant permis d'arriver à votre réponse.
2. Inversement, comment l'entier 24 se représente-t-il en base 2 ? Expliquez le raisonnement ayant conduit à votre réponse.

Exercice 2

1. Expliquez ce que fait le programme suivant, instruction par instruction. En particulier, après chaque instruction, donnez la valeur de `x` et celle de `y`.

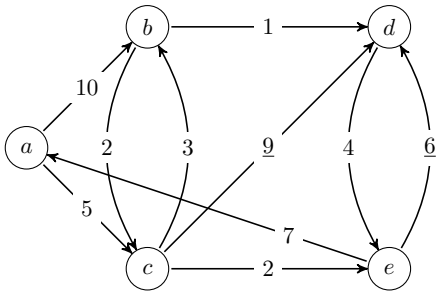
```
x = 3
y = 0
for i in range(x):
    y = y + 2
x = y
```

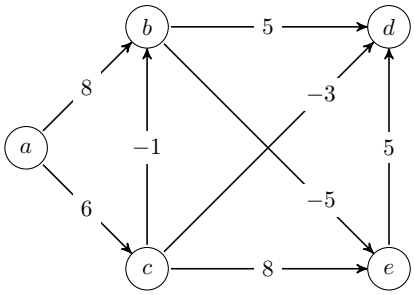
2. Plus généralement, on suppose que `x` a une valeur de départ `A` et que `y` a une valeur de départ `0`. Donner la valeur finale de `x` en fonction de `A` après les instructions suivantes :

```
for i in range(x):
    y = y + 2
x = y
```

Exercice 3

Soit les graphes orientés G_1 et G_2 (avec une fonction de pondération sur les arcs) représentés ci-dessous.





Pour chacun de ces deux graphes, on souhaite calculer les plus courts chemins **à partir du sommet** a.

1. Dites quel algorithme il faut choisir entre Bellman et Dijkstra pour chacun de ces deux graphes en justifiant votre choix en quelques mots.
2. Choisissez l'un des deux graphes et calculez ses plus courts chemins. On donnera pour chaque étape les modifications de la plus courte distance depuis a. Donnez également la liste des arcs formant l'arborescence des plus courts chemins, dans l'ordre lexicographique (par exemple si vous gardez les arcs (a, b), (b, c), (a, d), l'ordre devra être $(a, b), (a, d), (b, c)$).

Exercice 4

Les instructions suivantes

```
for i in range(10):
    print('z', end='')
```

permettent d'afficher 10 caractères `z`, sans espace ni retour à la ligne entre eux :

```
zzzzzzzzzz
```

Le but de `end=''` est justement de ne pas effectuer de retour à la ligne entre l'affichage de deux caractères `z`.

1. Écrire des instructions permettant d'afficher :

```
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz
```

2. Écrire une fonction `ligne(c)` permettant d'afficher une ligne de `c` caractères `z`, où `c` est un nombre donné en paramètre. Par exemple `ligne(c=4)` affichera :

```
zzzz
```

3. Modifier la fonction de la question précédente pour pouvoir afficher une ligne composée de caractères précisés dans un deuxième paramètre de la fonction : `ligne2(c, carac)` affiche une ligne faite de `c` caractères `carac`. Par exemple, `ligne2(c=10, carac='/')` affichera :

```
//////////
```

4. En utilisant soit la fonction `ligne2`, soit deux boucles `for` imbriquées, écrire une fonction `rectangle(l, c, carac)` qui affiche un rectangle de `l` lignes et `c` colonnes fait de caractères `carac`. Par exemple, `rectangle(l=3, c=10, carac='/')` affichera :

```
//////////
//////////
//////////
```

et `rectangle(l=2, c=5, carac='a')` affichera :

```
aaaaa
aaaaa
```

5. Écrire une suite d'instructions permettant d'afficher le motif suivant :

```
*****
****
***
**
*
```

Vous pouvez ou non utiliser la fonction `ligne2`.

6. Écrire une fonction `triangle(n)` permettant d'afficher un triangle fait de caractères `*` dont la première ligne a `n` caractères, la deuxième en a `n-1`, la troisième `n-2`, etc. et la n-ème ligne a `1` caractère. Par exemple, le triangle de la question précédente pourra être créé par l'appel `triangle(n=5)`.

Indication. Utiliser à nouveau la fonction ligne2, ou deux boucles imbriquées.