Spring 2026

Programmation 2

TP 2 — *Lisez attentivement l'ensemble du document.* — *Nabil Mustafa*

Consignes.

(a) **L'objectif est d'apprendre la programmation en Python.**

Il n'y a pas besoin de se précipiter !

Prenez votre temps et terminez correctement tous les TP précédents avant de commencer avec ce TP.

(b) *Vous devriez lire attentivement les diapositives, pour réviser ce que nous avons fait en CM.*

(c) *Il est important de se concentrer et de lire le texte attentivement et lentement. Si quelque chose dans le texte n'est pas clair après l'avoir lu plusieurs fois, demandez au professeur.*

Tâche 1: Lec2-conditionnelles.ipynb

(a) Connectez-vous à `LINUX` avec votre compte utilisateur.

(b) Téléchargez le Jupyter Notebook fichier `Lec2-conditionnelles.ipynb`, disponible sur le site web du module.

(c) Pour le lire, ouvrez un `terminal` dans le *même dossier* que le fichier `Lec2-conditionnelles.ipynb`, et exécuter le command :

```
jupyter-notebook Lec2-conditionnelles.ipynb
```

(d) Pour exécuter un code Python dans le Jupyter Notebook, cliquez sur le code et appuyez sur "SHIFT + ENTER".

(e) **Lisez toutes les informations** dans cette Jupyter Notebook.

(f) **Faites tous les exercices** dans cette Jupyter Notebook.

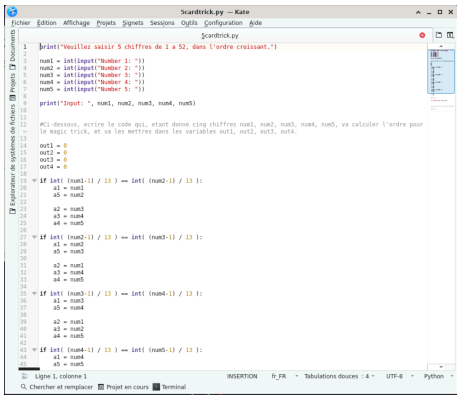
Tâche 2: Exécuter Python

(a) Connectez-vous à `LINUX` avec votre compte utilisateur.

(b) Téléchargez le fichier `5cardtrick.py`.

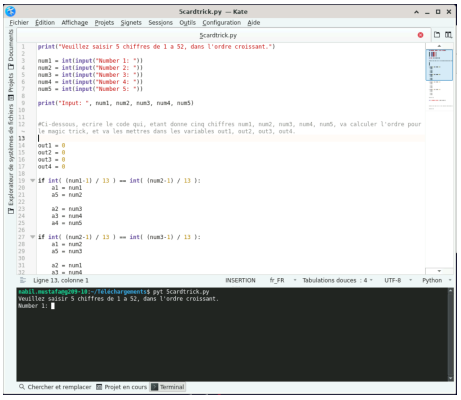
(c) Vous pouvez éditer ce fichier avec l'éditeur de votre choix. Par exemple, ouvrez un `terminal` dans le même dossier que le fichier `5cardtrick.py`, et exécuter le command:

```
kate 5cardtrick.py
```



(d) Pour exécuter le code avec `Python`, ouvrez un `terminal` dans le même dossier que le fichier `5cardtrick.py`, et exécuter le command :

```
python3 5cardtrick.py
```



Rappel: ce que nous avons fait en CM.

Étant donné 5 nombres distincts—chacun entre 1 et 52—**il y a un algorithme pour masquer l'un de ces 5 nombres et calculer un ordre pour les 4 nombres restants, de sorte qu'il soit possible de calculer le nombre qui est caché en examinant l'ordre des quatre nombres!**
Pour illustrer l'algorithme, prenons les 5 nombres

$$2 \quad 15 \quad 30 \quad 38 \quad 50$$

L'algorithme est le suivant:

a) Partitionnez *tous* les nombres entre 1 et 52 en quatre intervalles suivants:

$$[1-13] \qquad [14-26] \qquad [27-39] \qquad [40-52]$$

b) Étant donné les 5 nombres, choisissez deux nombres qui appartiennent au même intervalle.

Dans notre exemple, les nombres 30 et 38 appartiennent à l'intervalle $[27-39]$.

c) L'un de ces deux nombres deviendra le premier nombre dans notre ordre, et nous cacherons le deuxième.

Dans notre example, 30 ou 38 sera le premier nombre, et nous cacherons le deuxième.

d) Des deux nombres qui appartiennent au même intervalle, nous allons choisissez le premier nombre pour notre ordre, tel que la **distance circulaire** de ce premier nombre au deuxième nombre est inférieur ou égal à 6.

Dans notre example :
la distance circulaire $30 \longrightarrow 38$ est 8: $30 \to 31 \to 32 \to 33 \to 34 \to 35 \to 36 \to 37 \to 38$.
la distance circulaire $38 \longrightarrow 30$ est 5: $38 \to 39 \to 27 \to 28 \to 29 \to 30$.

Donc le premier nombre dans notre ordre va être 38 et on va cacher 30.

e) Nous encodons cette distance en permutant les 3 nombres restants. Disons que les trois nombres sont a, b et c en ordre croissant (donc $a < b < c$). On va mettre a, b et c dans un ordre comme suit. Je dispose de l'information suivante, une fois les trois autres nombres sont triés:

		2e	3e	4e
distance 1	$\Longrightarrow$	a	b	c
distance 2	$\Longrightarrow$	a	c	b
distance 3	$\Longrightarrow$	b	a	c
distance 4	$\Longrightarrow$	b	c	a
distance 5	$\Longrightarrow$	c	a	b
distance 6	$\Longrightarrow$	c	b	a

Dans notre example, les trois nombres restants sont 2 15 50. Donc $a = 2, b = 15, c = 50$. Il faut encoder la distance 5 avec l'ordre $c\ a\ b$, et donc l'ordre pour les trois nombres restants (a, b, c) est : 50 2 15.

f) Dans notre example, l'ordre finale est :

$$38 \quad 50 \quad 2 \quad 15.$$

et on va cacher le nombre 30.

Tour de magie: exemples.

Exemple 1:

a) Les 5 nombres sont :

$$4 \quad 12 \quad 26 \quad 28 \quad 46$$

b) 4 et 12 appartient à la même catégorie

c) Dans l'ordre circulaire, $12 \longrightarrow 4$ a distance 5 : $12 \to 13 \to 1 \to 2 \to 3 \to 4$.

d) On encode la distance 5 avec les 3 nombres restants, $a = 26, b = 28, c = 46$:

$$\text{distance } 5 \implies 46 \quad 26 \quad 28$$

Donc l'ordre est : $12 \quad 46 \quad 26 \quad 28$. Avec cet ordre, il est possible de calculer le 5eme nombre.

Exemple 2:

a) Les 5 nombres sont :

$$1 \quad 16 \quad 33 \quad 36 \quad 52$$

b) 36 et 33 appartient à la même catégorie

c) Dans l'ordre circulaire, $33 \longrightarrow 36$ a distance 3 : $33 \to 34 \to 35 \to 36$.

d) On encode la distance 3 avec les 3 nombres restants, $a = 1, b = 16, c = 52$:

$$\text{distance } 3 \implies 16 \quad 1 \quad 52$$

Donc l'ordre est : $33 \quad 16 \quad 1 \quad 52$. Avec cet ordre, il est possible de calculer le 5eme nombre.

Exemple 3:

a) Les 5 nombres sont :

$$7 \quad 18 \quad 30 \quad 40 \quad 51$$

b) 40 et 51 appartient à la même catégorie

c) Dans l'ordre circulaire, $51 \longrightarrow 40$ a distance 2 : $51 \to 52 \to 40$.

d) On encode la distance 3 avec les 3 nombres restants, $a = 7, b = 18, c = 30$:

$$\text{distance } 2 \implies 7 \quad 30 \quad 18$$

Donc l'ordre est : $51 \quad 7 \quad 30 \quad 18$. Avec cet ordre, il est possible de calculer le 5eme nombre.

Tâche 3: Encodage

Dans le fichier `5cardtrick.py`, ajouter le code qui

1. prend les 5 chiffres `num[0]`, `num[1]`, `num[2]`, `num[3]`, `num[4]`,
2. calcule l'ordre pour le magic trick, et les met dans les variables `output[0]`, `output[1]`, `output[2]`, `output[3]`.

Vous n'avez pas le droit d'utiliser de boucles.

Voici l'exemple :

```
$ python3 5cardtrick_sol.py
Veuillez saisir 5 chiffres de 1 a 52, dans l'ordre croissant.
Number 1: 5
Number 2: 18
Number 3: 29
Number 4: 37
Number 5: 40
Input:  5 18 29 37 40
Les quatre premiers chiffres sont :  37 40 5 18
5eme chiffre:  29
```

Indices.

Étape 1 : Identifier les deux nombres dans le même intervalle circulaire.

- Pour vérifier si `num[1]` et `num[2]` appartiennent au premier intervalle, on peut utiliser :

  ```
  if (num[1] >= 1 and num[1] <= 13 and num[2] >=1 and num[2] <= 13)
  ```

- En utilisant une logique similaire, vous pouvez vérifier si `num[1]` et `num[2]` appartiennent au deuxième intervalle, ou au troisième intervalle, ou au quatrième intervalle.
- Vous pouvez vérifier n'importe quelle paire de nombres de cette façon.

Étape 2 : Calculer la distance minimale sur le cercle.

- Soient x et y les deux nombres trouvés à l'étape 1, avec $x < y$.
- **Distance directe :** $d1 = y - x$. **Distance inverse :** $d2 = \cdots$?
- **Distance minimale :** $dmin = \min(d1, d2)$.

Étape 3 : Déterminer l'ordre d'émission.

- Si $dmin = d1$, alors on émet d'abord x (en position 1), puis on cache y en 5-éme.
- Si $dmin = d2$, alors on émet d'abord y (en position 1), puis on cache x en 5-éme.

Étape 4 : Permutation des trois nombres restants.

- Après avoir choisi quel nombre cacher, il reste trois nombres à transmettre.
- Maintenant, déterminez l'ordre de ces trois nombres aux positions 2, 3 et 4 en consultant la table d'encodage fournie précédemment, pour trouver la permutation correspondant à la distance $dmin$.

Tâche 4: Décodage

À la fin du même fichier `5cardtrick.py`, écrire le code qui prend les 4 premiers chiffres `output[0], output[1], output[2], output[3]`, et calcule le 5-eme chiffre `output[4]`.