Exercice 1 :
Soit le dictionnaire `roman = {"M":1000, "D":500, "C":100, "L":50}`

1) Compléter ce dictionnaire avec des entrées similaires pour les lettres X (10), V (5) et I (1).

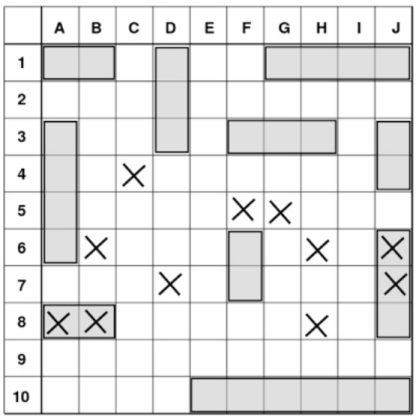
2) Utiliser le dictionnaire pour transformer un nombre écrit en chiffres romains en notation décimale.
Par exemple, le programme devra pouvoir transformer :

"MMXIX" en 2019

"DCCCXLIV" en 844.

3) Prolongation possible : écrire une fonction pour faire la transcription inverse.

Exercice 2 : Jeu de bataille navale



Utiliser un dictionnaire pour représenter les bateaux placés.
On pourra par exemple numéroter les bateaux :

```
sea = { }
sea["A1"] = 1
sea["B1"] = 1
sea["A3"] = 2
# etc.
```

Alternativement, on pourrait représenter les coordonnées par des couples de nombres naturels.
Au lieu de numéroter les bateaux, on pourrait leur attribuer des noms (torpilleur, croiseur...).
Par exemple :

```
sea = { }
sea[1,1] = "torpilleur"
sea[2,1] = "torpilleur"
sea[1,3] = "croiseur"
# etc.
```

Écrire un programme qui permet de placer un nombre fixe de bateaux.
Dans un premier temps, on se limitera à :

- un porte-avion : 5 cases
- deux croiseurs : 3 cases
- un torpilleur : 1 case

L'utilisateur pourra ensuite entrer des coordonnées pour essayer de faire couler les bateaux.
Le programme pourra utiliser l'opérateur **`in`** ou la méthode **`get()`** pour tester si un bateau a été touché et l'instruction **`del`** pour le faire couler.

Prolongations possibles :

- un jeu à un seul joueur, l'ordinateur place les bateaux aléatoirement
- un jeu à 2 joueurs, l'ordinateur place les bateaux dans 2 mers différentes
- un jeu à 2 joueurs, l'ordinateur place les bateaux dans une seule mer

(dans les 2 derniers cas, les joueurs ne savent pas où se trouvent leurs bateaux...)

Exercice 2 (plus détaillé) : Bataille navale :

Dans un premier temps, on se limitera à :

- un porte-avion : 5 cases
- deux croiseurs : 3 cases
- un torpilleur : 1 case

1) Établir la variable '**bateaux**', de type dict(), permettant de contenir les informations sur ces bateaux : nom (clé) et longueur (valeur).

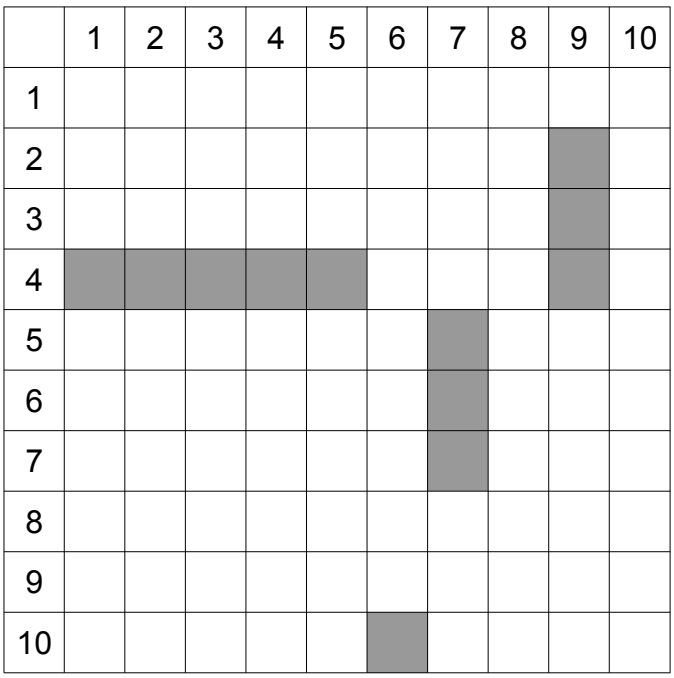
Le placement d'un bateau se fait horizontalement ou verticalement.
L'utilisateur précisera les cases de début et fin de chaque bateau.

Ces données seront enregistrées dans une variable '**mer**', de type dict(), avec comme :

clé : les cordonnées de la case occupée (tuple)
valeur : le nom du bateau

Exemple :

```
>>> mer
{(1, 4): 'porte-avion', (2, 4): 'porte-avion',
 (3, 4): 'porte-avion', (4, 4): 'porte-avion', (5, 4): 'porte-avion',
 (7, 5): 'croiseur_1', (7, 6): 'croiseur_1', (7, 7): 'croiseur_1',
 (9, 2): 'croiseur_2', (9, 3): 'croiseur_2', (9, 4): 'croiseur_2',
 (6, 10): 'torpilleur'}
```



Pour obtenir un dictionnaire référençant les positions des bateaux dans la mer, on s'assurera que les propositions faites par l'utilisateur sont valides : longueur respectée et cases libres.
Pour la suite, on pourra utiliser le dictionnaire '**mer**' ci-dessus, dans les tests de fonctions.

2) Écrire une fonction '**distance**', prenant en paramètre les coordonnées de 2 cases (2 tuples).
La fonction renvoie le nombre de cases entre les 2 (cases de départ et de fin comprises).

```
>>> distance((1, 4), (5, 4))
5
>>> distance((9, 2), (9, 4))
3
```

3) Écrire une fonction '**place_libre**', prenant en paramètre la mer (dict) et les coordonnées de 2 cases (2 tuples).
La fonction renvoie un booléen précisant si toutes les cases entre les 2 sont libres, cases de départ et de fin comprises *(utilisation de la méthode get()).*

```
>>> place _libre(mer, (3, 1), (3, 5))
False
>>> place_libre(mer, (8, 8), (10, 8))
True
```

4) Écrire une fonction '**placement_bateaux**', qui renvoie une mer (type dictionnaire) référençant les positions des bateaux contenus dans la liste '**liste_bateaux**'.
On demandera à l'utilisateur de proposer des positions qui devront être vérifiées par les fonctions '**distance**' et '**place_libre**'.

5) Écrire une fonction '**tir**', prenant en paramètre une mer (type dict) et les coordonnées d'une case (tuple). La fonction renvoie le résultat du tir (type str) dans cette case de la mer.

case vide : "A l'eau"

case occupée : Nom du bateau touché

```
>>> tir(mer, (3, 1))
A l'eau
>>> tir(mer, (7, 7))
croiseur_1
```

6) Écrire une fonction '**jeu_2_joueurs**', qui gère un jeu à deux joueurs :

Chaque joueur est invité à donner son nom, la position de ses bateaux.

A tour de rôle, les joueurs proposent un tir dans la mer adverse.

Chaque joueur est informé du résultat :

case vide : "A l'eau"

case occupée : Nom du bateau suivi de "touché" ou "coulé" (suivant le cas)

Si le tir est concluant, le joueur rejoue tout de suite.

Le gagnant est celui qui a coulé tous les bateaux de son adversaire.

Indication : A chaque case de bateau touché, on retirera celle-ci de la mer (fonction del). Un bateau n'étant plus référencé dans les valeurs d'un dictionnaire sera ainsi "coulé".

Prolongations possibles : voir bas de page 1