

## Sujet 7

### ❑ Exercice 1 : Minimum des éléments d'une liste

Ecrire une fonction `minimum` qui prend en argument une liste d'entiers *non vide* entiers et renvoie le minimum des éléments de cette liste.

Exemples :

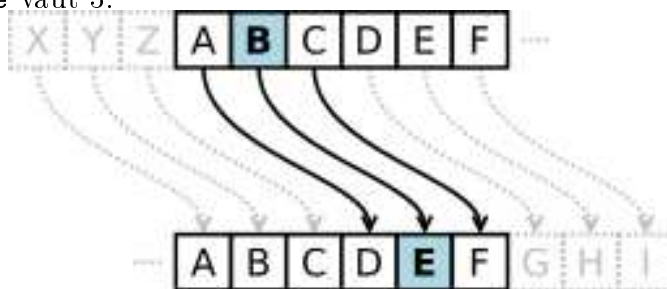
```

1 >>> minimum([12,11,8,20,15])
2 8
3 >>> minimum([5,-12,0,1,-4,7])
4 -12
5 >>> minimums([7,11,2])
6 2
7 >>> minimums([3])
8 3

```

### ❑ Exercice 2 : Codage de César

Etant donné un nombre entier `cle` compris entre 1 et 25, le chiffrement de César ou chiffrement par décalage consiste à décaler chaque lettre de `cle` emplacement dans l'alphabet. Par exemple dans l'illustration suivante `cle` vaut 3.



Avec un décalage de 3 on a donc : A qui devient D, B qui devient E, ..., X qui devient A, Y qui devient B, Z qui devient C. On veut écrire une fonction `chiffre` qui prend en argument une clé `cle` et un texte `texte` et renvoie ce texte chiffré avec le chiffrement de César. On suppose que le texte est *uniquement* constitué de lettres majuscules non accentuées. On propose l'algorithme suivant pour coder un caractère :

- Utiliser la fonction `ord` de Python pour obtenir son code
- Augmenter le code de `cle` et soustraire 26 si on dépasse le code de la lettre Z.
- Utiliser la fonction `chr` de Python pour obtenir la lettre codée.

```

1 def chiffre(texte,cle):
2     ''' Renvoie le chiffrement de texte par decalage de cle emplacements
3         '''
4     texte_code = ''
5     for caractere in texte:
6         num = ord(caractere)
7         num = num + cle
8         if num > ord('Z'):
9             num = num - 26
10    texte_code = texte_code + chr(num)
11    return texte_code

```

Exemples

```

1 >>> chiffre("NSI",3)
2 QVL
3 >>> chiffre("MATHS",12)
4 YMFTE

```