

2025 Asie jour 1

Exercice 1

1. $x = 0$ et $y = 20$.

2. Le code source d'un programme Python est du texte, il peut donc être stocké dans une chaîne de caractère.

3.

- programme3 : se termine
- programme4 : ne se termine pas
- programme5 : se termine
- programme6 : ne se termine pas

4. Il renvoie `True` si le programme s'arrête, par contre il ne renverra jamais `False` si le programme ne s'arrête pas.

5. On place mot au début du texte et on regarde en allant de droite à gauche si les lettres correspondent. À la première lettre qui ne correspond pas on décale le mot vers la droite d'un nombre de caractères qui dépend de la lettre qui a échoué.

6.

```
def arret_essai2(programme):  
    return not recherche("while", programme)
```

7.

- un programme avec un appel récursif infini peut ne pas s'arrêter sans `while` :

```
def programme7():  
    return programme7()
```

- un programme peut s'arrêter avec un `while` comme le programme3.

8.

```
def terminaison_inverse(programme):  
    if arret(programme):  
        boucle_infinie()
```

9. Supposons que `programme_paradoxal` termine.

Alors `terminaison_inverse(programme_paradoxal)` ne termine pas par définition de `terminaison_inverse`.

Or `programme_paradoxal = "terminaison_inverse(programme_paradoxal)"`, il y a donc un paradoxe.

10. Tous les autres programmes étant parfaitement définis, on en déduit que c'est l'hypothèse de l'existence de `arret` qui est fausse.

11. Non, quelque-soit le langage, il est impossible d'écrire une telle fonction.

Exercice 2

Partie A

1. Il y a 10 caractères donc 10 octets et donc 80 bits.

2. 53 49 58 20 41 4E 41 4E 41 53

Partie B

3.

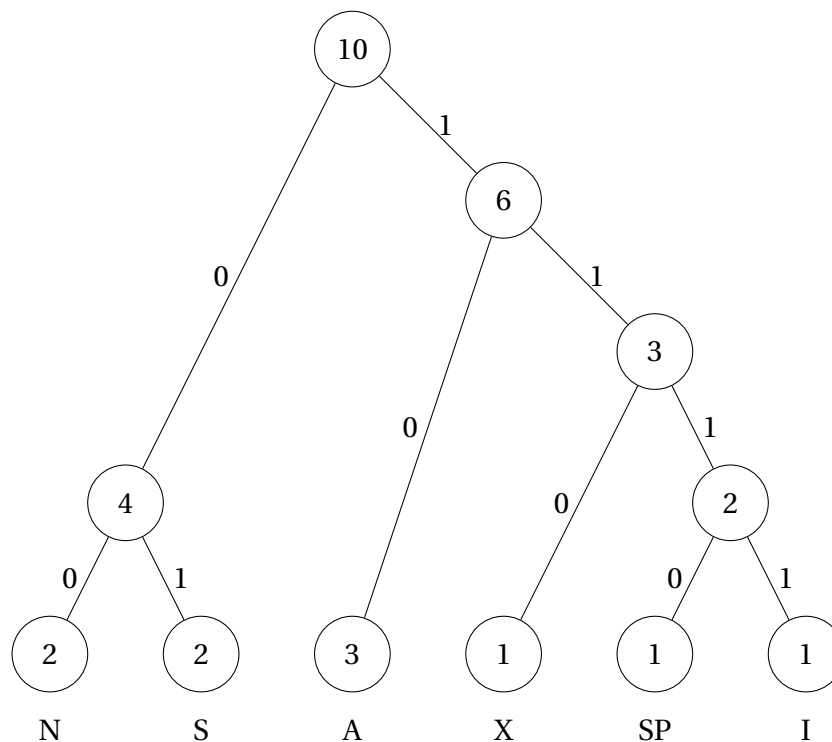
Symbole	SP	A	I	N	S	X
Nb occurrences	1	3	1	2	2	1

4. C'est le nombre de caractères.

5.

```
def occurrence(texte):  
    dico = {}  
    for lettre in texte :  
        if lettre in dico :  
            dico[lettre] = dico[lettre]+1  
        else:  
            dico[lettre] = 1  
    return dico
```

6. Voici un arbre possible :



Il y a d'autres arbres possibles car plusieurs nœuds sont identiques et le choix entre gauche ou droite est arbitraire.

7. Le poids de la racine correspond au nombre de caractères.

8. Un parcours en profondeur préfixe.

9.

Symbole	SP	A	I	N	S	X
code	1110	10	1111	00	01	110

10. Plus la lettre est fréquente, plus son code est court. Dans notre exemple il y a des codes sur deux, trois ou quatre bits.

11. 01 1111 110 1110 10 00 10 00 10 01

12.

$$t = \frac{80 - 25}{80} = 0.69 = 69\%$$

On a bien un nombre entre 20% et 90%.

Exercice 3

Partie A

1.

```
SELECT id-kimono
FROM location
WHERE fin = '';
```

2.

```
SELECT COUNT(id-kimono)
FROM kimono
WHERE taille = 130;
```

3.

```
SELECT nom, prenom
FROM adherent
JOIN location
ON adherent.numero-licence = location.numero-licence
WHERE id-kimono = 42 AND fin = '';
```

4.

```
UPDATE adherent
SET taille-adherent = taille-adherent + 10
WHERE taille-adherent < 160;
```

5. Il faut deux requêtes. On commence par le table `location` pour éviter des erreurs avec les clé étrangères.

```
DELETE FROM location
WHERE id-kimono = 25;
```

```
DELETE FROM kimono
WHERE id-kimono = 25;
```

Partie B

6. On admet que Eddie est un garçon : M12102021NIRRE01

7. La date de naissance est le 23/09/1974 et un nom de famille possible est MARTIN

8. 'STEPHANIE'

9. `tab_adherents[0]['numero-licence']`

10.

```
compteur = 0
for adherent in table:
    if adherent['annee'] == annee:
        compteur += 1
return compteur
```

11.

```
def adherent_plus_age(table):
    adherent_age = []
    annee = '2025'
    for adherent in table:
        if adherent['annee'] < annee:
            adherent_age = [adherent]
            annee = adherent['annee']
        elif adherent['annee'] == annee:
            adherent_age.append(adherent)
    return adherent_age
```

12. Il y a deux façons de faire.

```
def verification-licence(a):
    return a['sexe'] + a['jour'] + a['mois'] + a['annee']
        + extraire(a['nom'], 0, 5) == a['numero-licence']

def verification-licence(a):
    nl = a['numero-licence']
    return a['sexe'] == extraire(nl, 0, 1)
        AND a['jour'] == extraire(nl, 1, 3)
        AND a['mois'] == extraire(nl, 3, 5)
        AND a['annee'] == extraire(nl, 5, 9)
        AND extraire(a['nom'], 0, 5) == extraire(nl, 9, 14)
```