Lycée Les 3 Sources **Bourg-Lès-Valence**
SNT Année **2025-26**

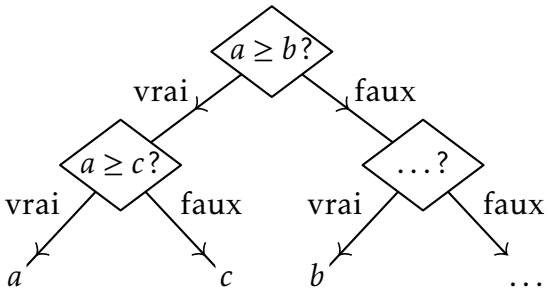
Tests et boucles non bornées – Feuille à rendre

Nom et prénom :

Compléter la feuille au fur et à mesure de votre avancement sur le site d'activités.

Exercice 1 :

Compléter l'arbre de décision ci-contre qui permet de traiter tous les cas.




```
def maximum3(a, b, c):
    if a >= b:
        if a >= c:
            return a
        else:
            return c
    else:
        if . . . . . .:
            return b
        else:
            return . . .
```

Exercice 2 :

Compléter la fonction ci-contre avec ce que vous avez mis dans votre fichier, en vous aidant de l'arbre au dessus.

```
def max3(a, b, c):
    if a >= b and a >= c:
        return a
    elif . . . . . and . . . . . .:
        return b
    else:
        return . . .
```

Exercice 3 :

Compléter la fonction ci-contre avec ce que vous avez mis dans votre fichier.

Exercice 4 :

Compléter la fonction suivante :

```
def demande_joueur():
    rep = . . . . . .
    while . . . . . .:
        rep = input("p/f/c ")
    return rep
```

Exercice 5 :

Le tableau ci-contre permet de déterminer qui gagne la manche dans les différents cas. v1 et v2 désignent les choix des deux joueurs. Remplir les cases avec 1 (si le joueur 1 gagne), 2 (si le joueur 2 gagne) et 0 en cas d'égalité.

Par exemple, le 0 dans le tableau signifie que si le joueur 1 choisit "p" (pierre) et que le joueur 2 aussi, alors il y a égalité. De même, le 1 signifie que si le joueur 1 choisit "f" et le joueur 2 propose "p", c'est le joueur 1 qui gagne.

v1 \ v2	"p"	"f"	"c"
"p"	0		
"f"	1		
"c"			

Exercice 6 :

Compléter la fonction ci-contre :

```
def determine_gagnant(v1, v2):
    if v1 == v2:
        return . . .
    elif v1 == "p" and v2 == "c":
        return 1
    elif v1 == "f" and v2 == . . .:
        return 1
    elif v1 == . . . and v2 == . . .:
        return 1
    else: # les autres cas
        return . . .
```

Exercice 7 :

Compléter la fonction ci-contre :

```
def manche():
    gagnant = 0
    while gagnant == . . .:
        v1 = demande_joueur()
        v2 = choice("pfc")
        print("L'ordinateur a choisi", v2)
        gagnant = determine_gagnant(. . ., . . .)
    if gagnant == . . .:
        print("Bravo, vous avez gagné")
    else:
        print("Vous avez perdu")
    return gagnant
```

Exercice 8 :

1) a) Si `score1` = 1 et `score2` = 3, combien est-ce qu'il y a eu de manches jouées? ...
   b) En utilisant `score1`, `score2` et `nb_manches`, donner la condition à mettre après **`while`** pour que la partie continue tant qu'il n'y a pas eu assez de manches de jouées. .....
2) a) À l'aide de `score1` et `score2`, donner une condition qui est vraie si le joueur 1 a gagné. .....
   b) Donner l'expression qui permet de savoir que le joueur 2 a gagné. .....
3) Compléter cette partie de la fonction :

```
if gagnant == . . .:
    score1 = score1 + . . .
else:
    score2 = . . . . . .
```

Exercice 9 :

On suppose que l'humain joue comme indiqué par les études statistiques.

1) Pourquoi est-ce que l'ordinateur a intérêt à jouer pierre au premier coup de la première manche? .....
   .....
2) Si le joueur 1 a fait pierre et l'ordinateur ciseaux, que doit jouer l'ordinateur à la prochaine manche pour avoir plus de chance de battre l'humain? .....
3) Si le joueur 1 a fait feuille et l'ordinateur ciseaux, que doit jouer l'ordinateur à la prochaine manche pour avoir plus de chance de battre l'humain? .....
4) Donner la séquence de symboles à donner pour gagner 5 manches consécutives contre l'ordinateur. Il ne doit pas y avoir d'égalité pendant une manche. .....