

## 2025 sujet 31

**Exercice 1**

```
# On utilise l'algorithme naïf

def recherche_motif(motif, texte):
    n = len(texte)
    m = len(motif)
    tab = []
    for i in range(n - m + 1):
        c = 0
        for j in range(m):
            if texte[i + j] == motif[j]:
                c += 1
        if c == m:
            tab.append(i)
    return tab
```

**Exercice 2**

```
def parcours(adj, x, acc):
    '''Réalise un parcours en profondeur récursif
    du graphe donné par les listes d'adjacence adj
    depuis le sommet x en accumulant les sommets
    rencontrés dans acc'''
    if x not in acc:
        acc.append(x)
        for y in adj[x]:
            parcours(adj, y, acc)

def accessibles(adj, x):
    '''Renvoie la liste des sommets accessibles dans le
    graphe donné par les listes d'adjacence adj depuis
    le sommet x.'''
    acc = []
    parcours(adj, x, acc)
    return acc
```