

## TD 8 - Informatique fondamentale

### Exercice 1

Proposer un code qui effectue les opérations suivantes, et calculer leur complexité.

- Afficher si un nombre  $n$  est pair ou impair
- Afficher un rectangle d'étoiles de largeur 5 et de longueur  $n$
- Afficher un carré d'étoiles de côté  $n$

### Exercice 2

Quelle est la complexité du code suivant ? On considérera  $n$  comme facteur le plus important, et l'instruction `append()` comme seule instruction coûteuse.

```
def maFonction(n):
    lines=[[1],[1,1]]
    for i in range(2,n):
        taille = len(lines[-1])
        newLine = [1]
        for i in range(taille-1):
            newLine.append(lines[-1][i]+lines[-1][i+1])
        newLine.append(1)
        lines.append(newLine)
    for j in range(len(lines)):
        print(lines[j])
    return 1
```

### Exercice 3

Quelle est la complexité du code suivant ? On considérera  $n$  comme facteur le plus important, et toute forme d'instruction comme coûteuse (comparaison `==`, assignation `=`, opération `%` et `+`, `append()`...).

```
#finding n prime numbers
def getNPrimes(n)
    firstPrimes = [1,2]
    foundPrimes = 2
    current = 3
    currentCheck = 1
    currentMaybePrime = True
    while foundPrimes < n:
        currentMaybePrime = True
        for i in range(2,current):
            currentCheck = current%i
            if currentCheck == 0:
                currentMaybePrime = False
        if currentMaybePrime:
            firstPrimes.append(current)
            foundPrimes = foundPrimes +1
        current = current+1
        currentCheck = 1
    print(firstPrimes)
```