

# Programmation Imperative

2024-2025

## Travaux Pratiques 3 (Structures de contrôle)

---

### Exercice 1. *ASCII*

Ecrire un programme qui affiche les caractères de "A" jusqu'à "Z" et de "0" jusqu'à "9" suivis la valeur de leur code ASCII.

*Exemple :*

```
caractere A code 65
```

### Exercice 2. *Menu*

Ecrire un programme proposant des choix des couleurs à l'aide des valeurs alphabétiques à l'utilisateur. Le programme affiche uniquement la couleur choisie par l'utilisateur, un message d'erreur est affiché si le choix n'appartient pas à l'ensemble des choix possibles.

*Les options de couleurs :*

```
printf ("b : bleu");  
printf ("n : noir");  
printf ("r : rouge ");  
printf ("v : vert");
```

Ajouter un choix de sortie et réafficher le menu (utiliser une fonction d'affichage) tant que le choix de sortie n'a pas été sélectionné.

*L'option de quitter :*

```
printf("q : quitter");
```

### Exercice 3. *Nombre narcissique*

Un nombre narcissique (ou en anglais PPDI, pour pluperfect digit invariant) est un entier naturel  $n$  non nul qui est égal à la somme des puissances  $p$ -ièmes de ses chiffres en base dix, où  $p$  désigne le nombre de chiffres de  $n$ .

*Exemples :*

- Tous les entiers de 1 à 9 sont narcissiques.
- 153 est un nombre narcissique :  
 $n = 153 \Rightarrow p = 3$  (153 a 3 chiffres)  
 $153 = 1^3 + 5^3 + 3^3$ .
- 93084 est un nombre narcissique :  
 $n = 93084 \Rightarrow p = 5$  (93084 a 5 chiffres)  
 $93084 = 9^5 + 3^5 + 0^5 + 8^5 + 4^5$ .

Ecrire un programme qui vérifie si un nombre est narcissique.

1ère étape : Trouvez combien de chiffres il y a dans le nombre  $n$  donné par l'utilisateur (c'est votre  $p$ ).

2ème étape : Multipliez chaque chiffre  $p$  fois et additionnez-les, **sans** utiliser la fonction `pow()`.

3ème étape : Vérifier si le résultat calculé dans l'étape 2 est égal à votre nombre initial  $n$ . Si oui, c'est un nombre narcissique, sinon, ce n'en est pas un.

