

# 11 Simulation numérique

- **Buts**

- Exercer les structures de contrôle et initialisation à la simulation numérique.

- **Travail à réaliser**

- Exercice 3.32 : Estimation de  $\pi$ . La bibliothèque `<random>` met à disposition divers outils pour la génération de nombres aléatoires. Le programme suivant donne des exemples simples de leur utilisation. Notez bien que chaque fonction de génération d'un nombre pseudo-aléatoire ne doit être instanciée qu'une et une seule fois.

```
#include <iostream>
#include <random>
#include <functional>
auto gen_double01 = std::bind(std::uniform_real_distribution<double>(0,1), std::mt19937(1234)); // germe: 1234
auto gen_int1_3 = std::bind(std::uniform_int_distribution<int>(1,3), std::mt19937(987)); // un autre germe
int main()
{
    std::cout << gen_double01() << " " << gen_int1_3() << std::endl;
}
```

- Exercice 3.33 : Le problème des 3 portes

- **Délai**

- Fin de la séance

# 12 Boucles `for` imbriquées

- **Buts**

- Exercer les structures de contrôle `for`

- **Travail à réaliser**

- Écrire un programme qui imprime des carrés imbriqués avec les lettres de l'alphabet. Par exemple, si l'utilisateur demande l'impression de 4 carrés, on aura l'impression suivante à l'écran :

```
DDDDDDDD
```

```
DCCCCCD
```

```
DCBBBB CD
```

```
DCBABCD
```

```
DCBBBB CD
```

```
DCCCCCD
```

```
DDDDDDDD
```

- Le programme doit être implanté sans utiliser de tableaux.

- **Délai**

- Fin de la séance

# 13 Suite de Syracuse

- **Buts**

- Exercer la conception de fonctions

- **Travail à réaliser**

- Exercice 4.16a : Écrire une fonction qui imprime le nombre et les valeurs successives prises par la suite de Syracuse pour une valeur entière donnée en paramètre. Essayer de concevoir une fonction qui puisse être réutilisée pour la seconde partie.
- Exercice 4.16b : Modifier le programme pour n'afficher que le nombre de valeurs, et afficher ce dernier pour les  $n$  premiers entiers,  $n$  fixé par l'utilisateur.

- **Délai**

- Fin de la séance