

# Atelier 7 : aller plus loin

*Pour générer un nombre aléatoire, vous pouvez utiliser les lignes suivantes :*

```
Random rand = new Random();  
int x = rand.nextInt(20); // retourne un nombre entre 0 et 19
```

## Exercice 1

Ecrire un algorithme simulant un lancer de pièce et affichant le résultat.

## Exercice 2

Ecrire un algorithme permettant de jouer au “Devinez ce nombre” : l'utilisateur doit deviner un nombre tiré aléatoirement entre 1 et 100. Lorsque le joueur saisit un chiffre, le jeu lui signale si le nombre à trouver est plus petit, plus grand, ou s'il a trouvé le bon nombre. Au lancement de la partie, le joueur peut choisir de jouer en facile (7 essais), moyen (5 essais) ou difficile (3 essais). Si le joueur atteint le nombre d'essais sans trouver le nombre, le jeu annonce qu'il a perdu.

## Exercice 3

Ecrire un algorithme permettant au joueur de faire une partie de pierre-feuille-ciseaux contre l'ordinateur.

## Exercice 4 :

Quelle boucle vous semble la plus adaptée pour le problème suivant ? Ecrivez l'algorithme correspondant.

L'utilisateur saisit un nombre. Ce nombre est divisé par deux jusqu'à valoir un, puis, on retourne le nombre de tours de boucles effectuées.

## Exercice 5 :

Quelle boucle vous semble la plus adaptée pour le problème suivant ? Ecrivez l'algorithme correspondant.

Je souhaite afficher tous les éléments pairs d'un tableau d'entiers

## Exercice 6 :

Quelle boucle vous semble la plus adaptée pour le problème suivant ? Ecrivez l'algorithme correspondant.

Un utilisateur doit saisir une donnée entre 0 et 20. Si jamais la valeur saisie n'est pas comprise dans cet intervalle, on redemande à l'utilisateur de saisir sa valeur.

### Exercice 7

Ecrire un algo qui demande à l'utilisateur de saisir un nombre. Puis, on affichera une grille de la taille du nombre saisie, remplie de X sur la diagonale basse et de O sur la diagonale haute. Par exemple, si l'utilisateur saisit 5, on voudra afficher :

```
X   O   O   O   O
X   X   O   O   O
X   X   X   O   O
X   X   X   X   O
X   X   X   X   X
```

Pour écrire un caractère sans retourner à la ligne, on peut utiliser `System.out.print()`.

### Exercice 8 : variante de la calculatrice

Ecrivez un algorithme affectant la valeur zéro à une variable. Puis, demander à l'utilisateur quelle opération il souhaite appliquer à cette variable. Une fois l'opération choisie, l'utilisateur doit saisir le deuxième nombre impliqué dans le calcul. Puis, le menu s'affiche de nouveau, permettant à l'utilisateur d'effectuer de nouvelles opérations.

Le menu doit également mettre à disposition une option pour indiquer qu'il a terminé.

## Exercice 9 (difficile)

Le blackjack est un jeu de cartes où l'objectif est d'obtenir un score le plus près possible de 21. Le joueur affronte un croupier : c'est celui qui a la valeur la plus proche de 21 qui l'emporte. Une partie se déroule de la manière suivante :

1. Le croupier tire une carte face visible, et une carte face cachée
2. Le croupier distribue deux cartes au joueur
3. Le joueur fait la somme de ses deux cartes : si il estime être suffisamment proche de 21, il peut choisir d'arrêter là. Sinon, il peut demander une nouvelle carte au croupier, autant de fois qu'il le veut. En revanche, si la somme dépasse 21, le joueur perd immédiatement.
4. Le croupier joue de manière similaire, mais a une règle simple : dès que la somme de ses cartes atteint 17 ou plus, il arrête de tirer. Toutes les cartes tirées après la première sont face cachée.
5. Une fois que le joueur et le croupier ont tous les deux cessé de tirer des cartes, ils comparent leur résultat : celui dont la valeur est la plus proche de 21 a gagné.

Les cartes ont les valeurs suivantes :

- De 2 à 10 : les cartes ont la valeur indiquée
- Les têtes (valet, dame, roi) valent chacun 10
- L'as peut prendre 2 valeurs, au choix de celui qui le possède : 1 ou 11. C'est lorsqu'il révèle les cartes que le joueur ou le croupier annonce la valeur choisie

Proposez un algorithme permettant de jouer au blackjack contre l'ordinateur, qui fera office de croupier.

## Exercice 10 (difficile)

Il existe des suites mathématiques qui se calculent sur plusieurs lignes. Dans cet exercice, on souhaite proposer un algorithme capable de créer le triangle de Pascal :

1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1

La formule pour la calculer est telle que représentée sur le schéma suivant :

0	+	1	+	0						
		v		v						
0	+	1	+	1	+	0				
		v		v		v				
0	+	1	+	2	+	1	+	0		
		v		v		v		v		
0	+	1	+	3	+	3	+	1	+	0
		v		v		v		v		v
0	+	1	+	4	+	6	+	4	+	1

Proposez un algorithme permettant de calculer cette suite. La ligne à laquelle on s'arrête est saisie par l'utilisateur.