

## Exercices

La factorielle d'un nombre entier positif  $n$ , notée  $n!$ , est le produit de tous les entiers positifs inférieurs ou égaux à  $n$ . Mathématiquement, cela s'exprime comme suit :

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$$

Par convention, on définit  $0! = 1$ . Ainsi, la factorielle d'un nombre  $n$  est le produit de tous les entiers positifs jusqu'à  $n$ , et le résultat est toujours un entier positif.

Quelques exemples de factorielles :

- $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
- $3! = 3 \times 2 \times 1 = 6$
- 1. •  $0! = 1$

Écrivez une fonction factorielle qui prend un nombre entier en paramètre et qui retourne la factorielle de ce nombre. Vous devez faire cet exercice d'abord avec la boucle `for`, ensuite la boucle `while` et à la fin la boucle `do ...while`. Votre fonction ou méthode ne doit pas utiliser `System.out.println`, elle doit retourner le résultat avec le mot-clé `return`

2. Écrivez une fonction qui demande à l'utilisateur de saisir une phrase et qui va afficher à l'écran une phrase cryptée de la façon suivante : Tous les caractères **a (en minuscule)** doivent être remplacés par le chiffre 5, les caractères **o (en minuscule)** par la lettre **v** et les caractères **e** par la lettre **B (majuscule)**
3. Chiffrement: Écrivez une fonction qui va crypter une phrase selon un décalage quelconque.
  - a. Votre fonction demande à l'utilisateur de saisir d'abord une phrase à chiffrer.
  - b. Ensuite, demander à l'utilisateur de saisir un nombre entre 1 et 25. Vous devez vous assurer que l'utilisateur saisit un nombre entier entre 1 et 25. Si l'utilisateur saisit une valeur qui n'est pas un nombre entier entre 1 et 25, vous devez afficher le message : « Vous devez afficher un nombre entier entre 1 et 25 » et vous devez terminer le programme. Ce qui veut dire que vous devez

aussi gérer le cas où l'utilisateur ne saisirait pas un nombre entier. Utilisez la méthode `hasNextInt()` pour vérifier que la saisie de l'utilisateur est vraiment un nombre entier.

- c. Si le nombre est bien un nombre entier compris entre 1 et 25, vous allez utiliser ce nombre pour décaler chaque caractère de la chaîne de caractère. Vous devez prendre en compte les majuscules et les minuscules. Les espaces et les nombres se trouvant dans la chaîne de caractère ne doivent pas être décalés.

Ex : `String phrase = Bonjour tout le monde;` `int decalage = 2;`

Résultat : `Dqpkpwt vqpv kp ogpvgp`

Il faut faire attention par exemple si vous arrivez à la lettre 'z' vous devez recommencer à la lettre 'a'.

Ex : `String phrase = Zidane;` `int decalage = 5`

Résultat : `Enidfs`