

Série de TD n°2

Exercice 1

1. Écrire la fonction factorielle `factNR` de manière non-réursive.
2. Écrire la fonction factorielle `factR` de manière réursive.
3. Simuler l'état de la pile lorsque l'instruction

```
res = factNR(4);
```

est exécutée.

4. Simuler l'état de la pile lorsque l'instruction

```
res = factR(4);
```

est exécutée.

Exercice 2

Écrire une fonction calculant la combinaison C_n^p en utilisant l'une des fonctions factorielles écrites à l'exercice 1. On rappelle que $C_n^p = \frac{n!}{(n-p)!p!}$.

Exercice 3

1. Écrire une fonction qui affiche la $i^{\text{ème}}$ ligne du triangle de Pascal.
2. Écrire une fonction qui appelle la fonction précédente pour afficher le triangle de Pascal jusqu'à une valeur entière n saisie au clavier.

Exercice 4

1. Écrire un algorithme qui teste si un nombre entier saisi au clavier est un nombre premier ou pas. On rappelle qu'un nombre entier est premier s'il est seulement divisible par 1 et par lui-même.
2. Écrire une fonction `testeNBPremier` correspondant à la traduction en langage C de l'algorithme donné à la question précédente.
3. Écrire un programme C permettant de trouver tous les nombres premiers compris entre deux nombres n_1 et n_2 saisis au clavier. (On vérifiera lors de la saisie que l'on a bien $0 < n_1 < n_2$.)