

Evaluation (1h)

Calcul sur les combinaisons et le triangle de Pascal

Q1 (4 pts) : Ecrire une fonction itérative (avec une boucle `while` ou bien une boucle `for`) permettant de calculer la fonction factorielle ($n \mapsto n!$). Cette fonction aura l'interface suivante :

```
int fact_iter(int n)
```

Q2 (4 pts) : En déduire une opération de calcul de C_n^p utilisant la définition suivante :

$$C_n^p = \frac{n!}{(n-p)! p!}$$

Cette fonction aura l'interface suivante :

```
int C_direct(int n, int p)
```

Q2bis (2pts) : Que pensez-vous de l'efficacité du calcul de la question 2 ? Proposez (sans écrire de programme) une solution pour l'optimiser.

Q3 (4 pts) : Utiliser la formule mathématique

$$C_{n+1}^{p+1} = C_n^{p+1} + C_n^p$$

pour programmer directement une fonction récursive calculant C_n^p à partir des valeurs initiales $C_n^0 = 1$ et $C_n^n = 1$. Cette fonction aura l'interface suivante :

```
int C_rec(int n, int p)
```

Q3bis (2 pts) : Que pensez-vous de l'efficacité du calcul de la question 3 ? Proposez (sans écrire de programme) une solution pour l'optimiser.

Q4 (4 pts) : Créer un programme calculant les coefficients d'une ligne donnée du triangle de Pascal. Cette fonction aura l'interface suivante et retournera un tableau à allouer dynamiquement et dont la taille sera adaptée au résultat à produire :

```
int* triangle(int ligne)
```

On utilisera la fonction `C_rec` pour calculer les coefficients binomiaux nécessaires. Pour mémoire, on rappelle la forme du triangle de Pascal :

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