

Programmation 2

Feuille de TD-TP n°3

Réversivité terminale et non terminale, mémorisation

Exercice 1 (TD). Calcul des coefficients binomiaux

Le but de cet exercice est calculer de plusieurs façons les nombres $\binom{n}{p}$, appelés *coefficients binomiaux*. Si $n, p \geq 0$, $\binom{n}{p}$ est égal au nombre de parties de p éléments d'un ensemble de n éléments. Les coefficients binomiaux vérifient le schéma de récurrence

$$\binom{n}{p} = \begin{cases} 0 & \text{si } p > n \\ 1 & \text{si } p = 0 \text{ ou } p = n \\ \binom{n-1}{p} + \binom{n-1}{p-1} & \text{si } 0 < p < n \end{cases}$$

Question 1.

1. Écrivez la définition d'une fonction **réursive** dont la déclaration est

```
unsigned bin (unsigned n, unsigned p);
```

qui reçoit deux entiers positifs n et p en arguments et renvoie la valeur de $\binom{n}{p}$, calculée en implémentant « naïvement » le schéma de récurrence ci-dessus.

2. Soit $C_{\text{bin}}(n, p)$ le nombre total d'appels à la fonction `bin` exécutés quand on appelle initialement la fonction sur les arguments n et p . Donnez une expression de $C_{\text{bin}}(n, p)$ en fonction de n et de p , puis calculez $C_{\text{bin}}(10, 4)$.

Question 2. Pour éviter l'exécution d'appels redondants, on stocke dans un tableau à deux dimensions, appelé par exemple `bin_tab`, le résultat des appels récursifs au fur et à mesure de leur obtention. On met à jour `bin_tab[m][q]` avec la valeur renvoyée par l'appel sur les deux arguments m et q la première fois qu'elle est calculée; lors d'un appel ultérieur avec les mêmes arguments, on récupère la valeur stockée au lieu d'appeler récursivement la fonction. Cette technique, illustrée en cours avec la fonction de Fibonacci, est appelée *mémorisation*.

1. Écrivez la définition d'une fonction **réursive** `bin_mem_2`, qui implémente cette stratégie avec le **même schéma de récurrence** que `bin`, de sorte que l'appel `bin_mem(n, p)` à la fonction définie ci-dessous renvoie la valeur de $\binom{n}{p}$.

```
#define N 50
unsigned bin_mem (unsigned n, unsigned p) {
    unsigned bin_tab [N] [N] = {0};
    return bin_mem_2 (n, p, bin_tab );
}
```

2. Soit $C_{\text{mem}}(n, p)$ le nombre total d'appels à la fonction `bin_mem_2` exécutés quand on appelle initialement la fonction `bin_mem` sur les arguments n et p . Donnez une expression de $C_{\text{mem}}(n, p)$ en fonction de n et de p , puis calculez $C_{\text{mem}}(10, 4)$.

Question 3. Les coefficients binomiaux vérifient aussi le schéma de récurrence suivant.

$$\binom{n}{p} = \begin{cases} 0 & \text{si } p > n \\ 1 & \text{si } p = 0 \text{ ou } p = n \\ \frac{n}{p} \binom{n-1}{p-1} & \text{si } 0 < p < n \end{cases}$$

1. Écrivez la définition d'une fonction **réursive terminale** `bin_ter_2` telle que l'appel `bin_ter(n, p)` à la fonction définie ci-dessous renvoie la valeur de $\binom{n}{p}$.

```
unsigned bin_ter (unsigned n, unsigned p) {
    return bin_ter_2 (n, p, 1, 1);
}
```

Question 4. Pour $0 \leq p \leq n$, on a aussi l'équation

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

1. Écrivez la définition d'une fonction **itérative** `bin_iter` qui reçoit deux entiers positifs n et p en arguments et renvoie la valeur de $\binom{n}{p}$, calculée en utilisant l'équation ci-dessus.

Exercice 2 (TP). Programme calculant les coefficients binomiaux

Écrivez un programme auquel on passe deux arguments depuis la ligne de commande, représentant respectivement deux entiers positifs n et p , et qui calcule et affiche la valeur du coefficient binomial $\binom{n}{p}$ en appelant successivement les quatre fonctions `bin`, `bin_mem`, `bin_ter` et `bin_iter` de l'exercice 1.

Par exemple, si `prog_bin` est le nom de l'exécutable obtenu en compilant votre programme, voici ce à quoi devrait ressembler l'affichage dans le terminal quand vous lancez cet exécutable avec les arguments 20 et 7 (le symbole `$` est l'invite de commande du terminal).

```
$ ./prog_bin 20 7
binomial (20, 7) = 77520 [nbre d'appels = 155039] (récurif naïf)
binomial (20, 7) = 77520 [nbre d'appels = 183] (récurif avec mémoïsation)
binomial (20, 7) = 77520 [nbre d'appels = 8] (récurif terminal)
binomial (20, 7) = 77520 (itératif)
$
```

Consignes. Tenez compte des trois consignes suivantes pour écrire et tester votre programme.

1. Pour l'utilisation des paramètres d'entrée de la fonction `main` (passage d'arguments depuis la ligne de commande), reportez-vous aux diapos du cours du 31 janvier, postées sur Moodle.
2. Pour le calcul et l'affichage du nombre d'appels exécutés pour chaque fonction, reportez-vous à l'énoncé de l'exercice 3 de la feuille de TD-TP n°2.
3. Les fonctions `bin_ter` et `bin_iter` de l'exercice 1 calculent de très grandes valeurs entières intermédiaires, même pour des valeurs « petites » des arguments n et p . Pour calculer $\binom{30}{12}$, par exemple, le type `unsigned int` ne permet pas de stocker ces calculs intermédiaires ; utilisez le type `unsigned long`, voire le type `unsigned long long`.