

Programmation : TP 7

Objectifs : [C, Ocaml] Tableaux (allocation, matrices)

Documentation

Les modules de la librairie standard d'OCaml sont documentés sur <http://rafale>, lien Objective Caml, lien The standard Library (Part IV), lien Module Array.

La documentation des fonctions de la librairie C est accessible avec `man : man malloc`.

[OCaml] Crible d'Eratosthène

L'algorithme du crible d'Eratosthène permet de déterminer tous les nombre premiers compris entre 1 et un entier n donné.

Le principe consiste à parcourir les entiers dans l'ordre croissant : 2 est premier, on supprime donc tous les multiples de 2 entre 3 et n . 3 n'a pas été supprimé puisqu'il n'est pas un multiple de 2 : il est donc premier. On supprime ses multiples, et ainsi de suite...

1. Écrire l'algorithme du crible d'Eratosthène en utilisant un tableau de booléens initialisés à `true`. La « suppression » des multiples se fera ici simplement en leur affectant la valeur `false`. Utiliser ensuite cette fonction pour afficher les nombres premiers entre 1 et n .

[C] Triangle de Pascal

On rappelle que le triangle de Pascal est construit en calculant les coefficients du polynôme $(x + 1)^n$ pour des valeurs successives de n .

```

1
1 1
1 2 1
1 3 3 1
...

```

La formule de récurrence permettant de calculer les coefficients de $(x + 1)^n$ est :

$$C_n^p = C_{n-1}^p + C_{n-1}^{p-1} \quad C_n^0 = C_n^n = 1$$

Dans cet exercice, nous allons essayer d'exploiter le fait que lorsqu'on calcule la $n^{\text{ième}}$ ligne du triangle de Pascal, les termes C_{n-1}^p et C_{n-1}^{p-1} ont déjà été calculés à la $(n - 1)^{\text{ième}}$ itération. L'idée de base consiste à stocker les coefficients dans des tableaux afin d'éviter de les recalculer.

1. Écrire un programme qui :
 - calcule (sans les afficher) et stocke dans une matrice tous les coefficients binômiaux du triangle de Pascal jusqu'à une profondeur m ;
 - affiche les lignes du triangle de Pascal à partir de la ligne n (n et m étant des valeurs de votre choix).
2. La solution retenue dans la question précédente est coûteuse en mémoire. En effet, il n'est nécessaire de mémoriser que la ligne précédente pour calculer les coefficients d'une ligne donnée. Écrire maintenant une fonction affichant les termes du triangle de Pascal jusqu'à une profondeur m , en utilisant seulement deux tableaux et des boucles `for`.