



1 Problème

- 1 Écrire un programme qui propose, via un menu, les actions suivantes (les prototypes de fonctions/procédures vous sont donnés):
1. Affichage d'un sapin : `void affichageSapin(int n);`
 2. Rang de la factorielle : `int maxFact(int k);`
 3. Nombre d'Armstrong : `void estArmstrong(int n);`
 4. Affichage binaire : `void binaire(int n);`
 5. Coefficients binômiaux : `int coefBin(int n, int p);` □

Affichage d'un sapin

Cette procédure affiche un sapin en utilisant uniquement les caractères ascii. Tout d'abord, le haut du sapin se dessine par un triangle isocèle en utilisant uniquement les caractères '*' et ' ' (espace). Ensuite, le tronc est dessiné par un carré (3x3) de '@'. L'entier n permet de connaître la taille du haut du sapin. Voici un exemple pour $n = 5$:

```
      *
     ***
    *****
   ********
  *********
 *****
 @@@
 @@@
 @@@
```

Rang de la factorielle

Cette fonction doit calculer, pour un entier k donné, le plus grand entier n tel que $n! \leq k$.

Nombre d'Armstrong

Cette procédure doit définir si un nombre (saisi par l'utilisateur) est un nombre d'Armstrong. C'est-à-dire que ce nombre est égal à la somme des cubes des chiffres qui le composent.

Exemple : $153 : 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

Quelques nombres d'Armstrong : 1, 153, 370, 371, 407.

Affichage binaire

Cette procédure prend en paramètre un entier naturel n et affiche son écriture binaire (en base 2).

Coefficients binômiaux

Cette fonction calcule le coefficient binomial $C_n^p = \frac{n!}{(n-p)!p!}$.

Quelle est la complexité de cette fonction? Quelle est la valeur maximale calculable?

Proposer une (ou plusieurs) autre(s) méthode(s) (non récursives) moins coûteuse et permettant de calculer des valeurs de C_n^p beaucoup plus grandes.