

Objectifs du TD

Le but de ce TD est de vous faire découvrir la syntaxe propre au langage Java ainsi que de vous faire travailler sur les tableaux.

Triangle de Pascal

Le triangle de Pascal est une présentation des coefficients binomiaux dans un triangle. La construction du triangle est régie par la relation de Pascal (pour tout entier n et k tels que $0 < k < n$):

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Le coefficient de la ligne n et colonne k s'obtient en ajoutant les coefficients de la ligne $n-1$ et colonne $k-1$ et de la ligne $n-1$ et colonne k . De plus on a :

$$\binom{n}{0} = \binom{n}{n} = 1$$

Le dessin ci-dessous illustre les six premières lignes du triangle de Pascal :

$$\begin{array}{cccccc}
 & & 1 & & & \\
 & 1 & & 1 & & \\
 & 1 & 2 & & 1 & \\
 1 & 3 & 3 & & 1 & \\
 1 & 4 & 6 & 4 & 1 & \\
 1 & 5 & 10 & 10 & 5 & 1
 \end{array}
 \quad
 \begin{array}{l}
 \binom{3}{0}=1=\binom{3}{3} \\
 \binom{5}{1}=5=\binom{5}{4} \\
 \binom{5}{2}=10=\binom{5}{3}
 \end{array}$$

Questions

Il vous faut lire le code présent en page 2 pour pouvoir répondre aux questions.

Question 1 Que produit l'exécution du `main` de ce code ?

Question 2 Pourquoi pensez-vous que le type `long` a été utilisé pour les cases des tableaux alors que le type `int` est utilisé pour le nombre de ligne ?

Question 3 Que pensez-vous du code ? Quelles parties auriez-vous coder différemment ?

Question 4 Donner le code d'une fonction `long binomial(int n, int k)` qui renvoie $\binom{n}{k}$ en utilisant le triangle de Pascal

Question 5 Donner le code d'une fonction `void printPascalTriangle(long maxValue)` qui affiche les lignes du triangle de Pascal dont les valeurs sont inférieures ou égales à `maxValue`. Quelle fonction intermédiaire pourriez-vous coder pour réaliser cette fonction ?

Code Java

Afin d'afficher le triangle de Pascal on va utiliser le code Java suivant.

```
public class PascalTriangle {
    static long[] nextRow(long[] currentRow){
        long[] nextRow = new long[currentRow.length + 1];
        nextRow[0] = 1;
        nextRow[nextRow.length - 1] = 1;
        for(int column = 1; column < nextRow.length - 1; column++) {
            nextRow[column] = currentRow[column - 1] + currentRow[column];
        }
        return nextRow;
    }

    static void printRow(long[] row){
        for(long value : row){
            System.out.print(value + " ");
        }
        System.out.println();
    }

    static void printPascalTriangle(int numberOfRows){
        long[] currentRow = {1};
        for(int row = 0; row < numberOfRows; row++){
            printRow(currentRow);
            currentRow = nextRow(currentRow);
        }
    }

    public static void main(String[] args){
```

```
    printPascalTriangle(6);  
  }  
}
```