
Série N°3
TD Algorithmique & Programmation

Exercice 1

Définir les fonctions suivantes :

- **Triangulaire(int n)** : qui retourne 1 si l'entier n est triangulaire et 0 sinon.
Un nombre N est dit triangulaire s'il existe un $k > 0$ tel que : $N = 1 + 2 + \dots + k$
Exemple : 15 est un nombre triangulaire car $15 = 1+2+3+4+5$
- **Tetraedrique(int n)** : qui retourne p si l'entier n est la somme des p premiers entiers triangulaires (on dit que n est un tétraedrique d'ordre p) et 0 sinon.

Exercice 2

Le nombre de combinaisons de i dans n est défini par :

$$C_n^i = n! / (i! * (n - i)!)$$

1. Ecrire une fonction qui calcule le nombre de combinaisons de i dans n
2. Le triangle de Pascal est défini par :

$$\begin{array}{ccccccc}
 & & C_0^0 & & & & \\
 & & \vdots & & & & \\
 & C_1^0 & C_1^1 & & & & \\
 & \vdots & \vdots & & & & \\
 & C_2^0 & C_2^1 & C_2^2 & & & \\
 & \vdots & \vdots & \vdots & & & \\
 & \vdots & \vdots & \vdots & & & \\
 & C_n^0 & C_n^1 & C_n^i & \dots & C_n^n &
 \end{array}$$

Ecrire une fonction qui édite le triangle de Pascal en respectant la forme d'affichage donnée

Exercice 3

Donner le résultat d'exécution du programme suivant :

<pre>#include <stdio.h> void avancer1(int x){ x+=1; } int avancer2(int *x){ *x+=2; return *x ; } void reculer2(int x){ x-=2; }</pre>	<pre>int reculer3(int *x){ *x-=3; return *x ; } main(){ int x=6, y=1; avancer1(x); printf("\n x=%d",x); x=avancer2(&y); printf("\n y=%d",y); printf("\n x=%d",x);</pre>	<pre> reculer2(y); printf("\n y=%d",y); y=avancer2(&x); printf("\n y=%d",y); printf("\n x=%d",x); y=reculer3(&x); printf("\n y=%d",y); printf("\n x=%d",x); x=reculer3(&y); printf("\n x=%d",x); printf("\n y=%d",y); }</pre>
---	---	--

Exercice 4

1. Ecrire une fonction **Max_Min** à 3 arguments qui calcule le maximum et le minimum d'une série de n entiers saisis au clavier. Les variables relatives au minimum et au maximum font partie des arguments de la fonction : void **Max_Min**(int, int, int)
2. Ecrire un programme qui fait appel à la fonction **Max_Min** avec trois façons différentes.

<pre>int main() { int n, min, max //contrôle de saisie sur n Max_Min(..., ..., ...) ; printf("\nLe max = %d ", ...) ; printf("\nLe min = %d\n",....) ; }</pre>	<pre>int main() { int n, *min, *max ; //contrôle de saisie sur n Max_Min(..., ...,) printf("\nLe max = %d ",) ; printf("\nLe min = %d\n",....) ; }</pre>	<pre>int main() { int *n, *min, *max ; //contrôle de saisie sur n Max_Min(..., ...,) printf("\nLe max = %d ",) ; printf("\nLe min = %d\n",....) ; }</pre>
---	---	--

Exercice 5

La forme récursive du nombre de combinaisons de **i** dans **n** est définie par :

$$C_n^i = C_{n-1}^i + C_{n-1}^{i-1}$$

Ecrire une fonction récursive qui calcule le nombre de combinaisons de **i** dans **n**

Exercice 6

Etant donnée la suite V_n définie par :

$$\begin{cases} V_0 = 2 \\ V_n = V_{n-1} + 2 & \text{si } n \text{ pair} \\ V_n = V_{n-1} + 4 & \text{si } n \text{ impair} \end{cases}$$

1. Ecrire une fonction récursive qui permet de calculer le **n**^{ième} terme de la suite V_n .
2. Ecrire une fonction qui permet de calculer la somme des **N** premiers termes de la suite V_n .
3. Ecrire un programme qui permet de saisir un entier **N** > 0, puis calcule la somme des **N** premiers termes de la suite V_n .