

Langage C
TP5
Itheri Yahiaoui && Stéphane Cormier

Exercice 1 : « Allocation dynamique 1D »

- Ecrire un programme qui permet de
 - convertir un entier positif en binaire, (par exemple l'entier 10 donnera $(1010)_2$),
 - stocker la représentation binaire dans un tableau à raison d'un chiffre par case, la taille du tableau doit être strictement égale au nombre de chiffres présents dans la représentation binaire.
 - afficher la taille du tableau, son contenu, le nombre de zéros et le nombre des uns.
- Ecrire un programme qui demande à l'utilisateur une longueur pour créer deux tableaux dynamiques à une dimension. Chaque tableau représentera un vecteur de taille égale à la longueur saisie. Ces deux vecteurs seront initialisés par l'utilisateur avant le calcul des distances suivantes entre eux.
« [https://fr.wikipedia.org/wiki/Distance_\(math%C3%A9matiques\)](https://fr.wikipedia.org/wiki/Distance_(math%C3%A9matiques)) »:
 - la distance de Manhattan,
 - la distance euclidienne,
 - la distance de Minkowski,
 - la distance de Tchebychev.N'oublier pas de libérer la mémoire allouée avant la fin du programme.
- Ecrire un programme qui initialise deux tableaux de la même manière que la question précédente et qui calcule et stocke le résultat des opérations suivantes dans des structures adéquates :
 - la somme des deux vecteurs (somme des éléments deux à deux),
 - la soustraction des deux vecteurs (soustraction des éléments deux à deux),
 - le produit des deux vecteurs (produit des éléments deux à deux),
 - le produit scalaire des deux vecteurs.N'oublier pas de libérer la mémoire allouée avant la fin du programme.

Exercice 2 : « Allocation dynamique 2D »

- Ecrire un programme qui
 - demande à l'utilisateur de saisir la taille (en nombre de points) d'un nuage de points dans l'espace 3D,
 - initialise aléatoirement les coordonnées de ces points (la valeur de chaque coordonnée est supérieure ou égale à -1000 et inférieure à 1000),
 - stocke les coordonnées des points dans un tableau 2D créé dynamiquement de taille $N \times 3$.
 - affiche à l'utilisateur un menu avec les propositions suivantes :
 - a) Calculer le nombre de points se trouvant dans un octant « sous-espace ».
 - b) Calculer la moyenne des points se trouvant dans un octant « sous espace ».

- c) Copier les points se trouvant dans un octant dans une matrice appelée Points_Octant_i, i allant de 1 à 8.
 - d) Calculer la distance euclidienne entre deux points.
 - e) Vérifier si trois points forment un triangle, si c'est le cas, afficher son périmètre, et sa surface.
- Si l'utilisateur choisit le a) , b) ou c) un deuxième sous-menu est affiché pour le choix de l'octant : 1) (+, +, +), 2) (+, +, -), 3) (+,-,+), 4) (+,-,-), 5) (-,+,+), 6) (-,+,-), 7) (-, -, +), 8) (-, -, -) .
 - Si l'utilisateur choisit d), on lui demande les indices des deux points dans le tableau.
 - Si l'utilisateur choisit e), on lui demande les indices des trois points dans le tableau.
- D'après Wikipédia, la suite de Conway est une suite mathématique inventée en 1986 par le mathématicien John Horton Conway, initialement sous le nom de "suite audioactive". Elle est également connue sous le nom anglais de Look and Say ("regarder et dire"). Dans cette suite, un terme se détermine en annonçant les chiffres formant le terme précédent. Le premier terme de la suite de Conway est posé comme égal à 1. Chaque terme de la suite se construit en annonçant le terme précédent, c'est-à-dire en indiquant combien de fois chacun de ses chiffres se répète.

X0 = 1

X1 = 11

X2 = 21

X3 = 1211

X4 = 111221

X5 = 312211

X6 = 13112221

Xn=.....

- Ecrire un programme qui calcule et affiche les termes de la suite de Conway jusqu'à un certain rang n en utilisant un tableau statique.
- Ecrire un programme qui calcule et affiche les termes de la suite de Conway jusqu'à un certain rang n en utilisant un tableau dynamique qui n'utilise que la mémoire strictement nécessaire à son stockage.