

LIFAPI – Partie B – Langage C/C++

Contrôle Continu Terminal (Durée totale : 2h)

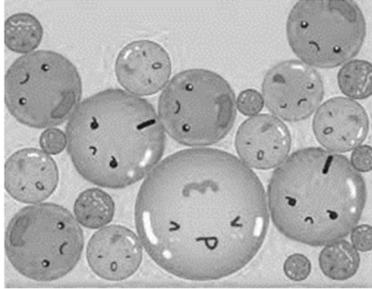
Lundi 18 décembre 2023

Recommandations : Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.

NOM :

PRENOM :

Numéro Etudiant :



Suika Game, parfois appelé le Jeu de la pastèque, est un jeu vidéo japonais de réflexion développé par Aladdin X. Le but de Suika Game est de faire tomber des fruits dans un récipient afin de les combiner jusqu'à l'obtention d'une pastèque (Suika signifie pastèque en japonais). Nous allons dans tout cet exercice effectuer une modélisation simplifiée du célèbre jeu. L'aspect graphique sera mis de côté et la plupart des sous-programmes non essentiels sur papier ne seront pas demandés.

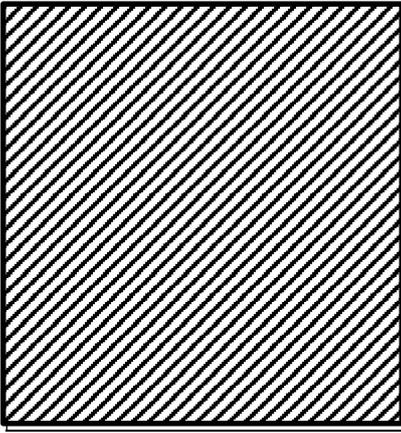
Préambule

- 1- Ecrire en C/C++ une **procédure** `val_alea` qui choisit **aléatoirement** et "retourne" une valeur aléatoire comprise entre `a` et `b` inclus (on supposera que `a < b` sans le vérifier).

Structures de données

- 2- Définir en C/C++ une constante entière `MAXCERCLES` ayant pour valeur 30.

- 3- Définir en C/C++ une structure `cercle` contenant les coordonnées (x_c, y_c) du centre et le rayon `r` du cercle.



4- Définir en C/C++ une structure `suika` contenant un entier `nb` et un tableau `tab` de taille maximale `MAXCERCLES` contenant des `cercle`.

5- Ecrire en C/C++ une **fonction** `creer_cercle` qui remplit toutes les caractéristiques d'un `cercle` et retourne le `cercle` généré. Le rayon `r` sera un entier compris entre 10 et 100 multiple de 10 (en utilisant la question 1-, on générera une valeur aléatoire entre 1 et 10 et on multipliera le résultat obtenu par 10), les coordonnées du centre (xc, yc) seront générées aléatoirement de manière à ce que l'intégralité du cercle se trouve dans une fenêtre de taille 500*500 (on prendra en compte la valeur du rayon).

6- Ecrire en C/C++ un sous-programme `afficher_cercle` qui affiche les caractéristiques d'un `cercle`.

7- Ecrire en C/C++ un sous-programme `ajouter_un_cercle` qui ajoute **un** cercle au tableau `tab` de la structure `suika` s'il reste de la place.

8- On peut déterminer s'il y a intersection entre deux cercles en calculant la distance D entre les deux centres (on utilisera le théorème de Pythagore) et en la comparant avec la somme S des rayons des deux cercles. Si $D < S$ alors il y a intersection, si $D = S$ alors ils sont tangents et sinon ils sont disjoints. Ecrire en C/C++ une fonction **booléenne** `intersection` qui retourne vrai si deux cercles `C1` et `C2` passés en paramètres s'intersectent ou sont tangents, faux sinon.

9- Dans le jeu `suika`, deux fruits (cercles) de même taille fusionnent pour en donner un plus gros s'ils s'intersectent. A chacune des étapes du jeu, le programme vérifiera s'il existe deux cercles qui remplissent ces conditions.

Ecrire en C/C++ une fonction booléenne `premiere_intersection` qui retourne vrai s'il existe deux cercles qui peuvent fusionner et faux sinon. La fonction "renverra" en plus les indices `ic1` et `ic2` des deux cercles du tableau qui seront à fusionner (ou -1 s'il n'y a pas d'intersection).

10- Ecrire en C/C++ un sous-programme `fusion` qui à partir d'un jeu `suika` et de deux indices `ic1` et `ic2` de cercles dans cette structure, effectue la fusion de ces deux cercles. Le cercle d'indice `ic1` sera remplacé par le cercle résultat de la fusion (nouveau centre = moyenne des positions des deux centres et rayon augmenté de 10). Le cercle d'indice `ic2` sera quant à lui remplacé par le dernier cercle du tableau.