LIFAP1 – Séquence 5

TP Noté #2 - durée 1h30 mn

Jeudi 16 décembre 2021

Sujet B

Consignes

Aucun accès au WEB, aux pages de l'UE, ni à vos anciens TP n'est autorisé. Dans votre fichier, vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant.
La note tiendra compte du respect des consignes, de la qualité de la présentation et de la lisibilité du code, des algorithmes, et du bon fonctionnement du programme. **Seules les notions vues en cours devront être utilisées**.
Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot_TP_NOTE de l'UE LIFAP1). Aucun retour par mail ne sera accepté.

Travail à réaliser

On souhaite développer une petite application mathématique permettant de gérer des polygones définis par des points du plan. Vous pourrez utiliser les fonctions de la bibliothèque `math.h`.

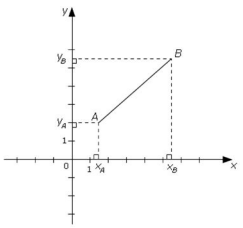
1. Définir en C/C++ une constante `MAX_POINTS` ayant pour valeur 10.
2. Un `point` est défini en 2D par son `abscisse` et son `ordonnee` (deux réels). Définir en C/C++ la structure `point`.
3. Écrire en langage C/C++ une **fonction** `remplir_point` permettant de remplir une structure `point` avec des informations choisies par l'utilisateur.
4. Ecrire en C/C++ une procédure `affiche_point` permettant d'afficher toutes les caractéristiques d'un `point` passé en paramètre.

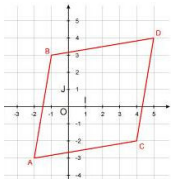
Un `polygone` est défini par un ensemble de `points` du plan. Définir en C/C++ la structure `polygone` contenant le nombre de `points` du polygone noté `nb_pts`, ainsi qu'un tableau de `points` appelé `tab_pts` ayant au maximum `MAX_POINTS` cases.

5. Écrire en langage C/C++ une **procédure** `ajoute_un_point` permettant d'ajouter un `point` au `polygone` s'il reste encore de la place. On fera appel à la procédure écrite en 3.
6. Écrire en langage C/C++ une procédure `affiche_polygone` qui en utilisant la procédure écrite en 4 va afficher les coordonnées de tous les `points` d'un `polygone` passé en paramètre.
7. Écrire en langage C/C++ une **fonction** `distance` qui calcule et retourne la distance entre 2 `points` passés en paramètres. On pourra utiliser la fonction `pow(x,y)` qui retourne la valeur de x^y et la fonction `sqrt(x)` qui retourne la racine carrée d'un nombre `x` passé en paramètre.

Le plan est muni d'un repère orthonormé.
Soient A et B deux points du plan, $(x_A ; y_A)$ et $(x_B ; y_B)$ leurs coordonnées. La distance de A à B est donnée par la formule suivante :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}.$$





8. Écrire en langage C/C++ un sous-programme `longueur_segments` permettant de remplir un tableau avec les longueurs de tous les segments composant le polygone. Dans l'exemple ci-contre, il faudra calculer les longueurs des segments AB / BC / CD et DA. Le `polygone p` et le tableau `tab_dist` à remplir seront passés en paramètres.
9. Écrire en langage C/C++ une fonction booléenne `toutes_identiques` qui retourne vrai si les `n` premières valeurs contenues dans le tableau `tab_dist` passé en paramètre sont identiques, faux sinon.

Cas `vrai`

3.14	3.14	3.14	3.14									

Cas `faux`

3.14	3.14	3.14	3.14	4.37	7.32							

10. Écrire en langage C/C++ le programme principal qui à partir des sous-programmes écrits précédemment permet
    a. de saisir les coordonnées de 4 points qui constitueront un polygone,
    b. de déterminer et d'afficher si ce polygone est un losange ou non. Pour rappel un losange est un quadrilatère dont les 4 cotés sont égaux. (Testez avec les valeurs de la figure ci-dessus).