

IF2B – Examen final

Durée : 1h30

Seule une feuille A4 manuscrite nominative recto-verso est autorisée comme document.

Tous les autres documents ou tout autre dispositif électronique sont interdits.

Barème donné à titre indicatif (± 1)

Remarque : vous pouvez rédiger les algorithmes demandés en pseudo-code ou en langage C (sauf si le langage C est imposé par l'exercice). Le prototype complet des sous-programmes doit être fourni. Il n'est pas nécessaire de préciser les « includes » pour les exercices en C. Il n'est pas nécessaire d'écrire de programme *main*.

Exercice 1 – Recherche de motif (7 points)

Nous souhaitons réaliser une fonction capable de rechercher un « sous-tableau 2D » donné (appelé « motif ») dans un plus grand tableau 2D (appelé « canvas »). Les deux tableaux contiennent des entiers. On les supposera toujours carrés (la largeur égale la hauteur).

Par exemple :

- Le motif

| | | |
|---|---|---|
| 5 | 3 | 5 |
| 1 | 3 | 4 |
| 8 | 1 | 2 |

- Le canvas

| | | | | | |
|---|---|---|---|---|---|
| 9 | 6 | 4 | 5 | 1 | 4 |
| 4 | 9 | 0 | 5 | 9 | 2 |
| 2 | 5 | 5 | 3 | 5 | 6 |
| 0 | 3 | 1 | 3 | 4 | 8 |
| 1 | 9 | 8 | 1 | 2 | 5 |
| 9 | 6 | 4 | 5 | 1 | 4 |

Dans l'exemple, le « motif » est bien présent dans le « canvas » et est surligné. On peut également dire que le motif se trouve à la position (2, 2) car il commence à la colonne et à la ligne d'indice 2.

L'objectif est d'écrire une fonction qui, étant donné un motif et ses dimensions, ainsi qu'un canvas et ses dimensions, retourne 1 si le motif se trouve dans le canvas et 0 sinon.

- 1) **Ecrire une 1^{ère} fonction, qui étant donné un motif, un canvas, leurs dimensions respectives, ainsi qu'une position dans le canvas, retourne 1 si le motif se trouve à cette position, et 0 sinon**
- 2) **Ecrire une 2^{ème} fonction, appelant la précédente, pour répondre à l'énoncé**

Exercice 2 – Polygones (13 points)

Nous disposons d'un fichier contenant les informations représentant des formes géométriques (polygones). Chaque ligne du fichier contient les informations d'un polygone. Chaque polygone est défini par un nombre de points (au minimum 1, au maximum 10) et des coordonnées 2D de ces points (deux réels pour chaque point).

Les données d'un polygone sont toutes séparées par des virgules, sous la forme :

« Nombre de point, coordonnée X du 1^{er} point, coordonnée Y du 1^{er} point, coordonnée X du 2^{ème} point, coordonnée Y du 2^{ème} point, ... »

Par exemple, un fichier contenant un triangle et un quadrilatère serait sous la forme :

3, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0

4, -1.0, -1.0, 1.0, -1.0, 1.0, 1.0, -1.0, 1.0

- 1) Définir en C, un type structure Point2D représentant un point 2D d'un polygone tel que défini dans cet énoncé et un type structure Polygone représentant un polygone

Pour les questions qui suivent vous pouvez utiliser les structures Point2D et Polygone préalablement définies.¹

- 2) Ecrire en C, une fonction qui, étant donné le chemin d'un fichier, un tableau de polygones (et sa taille), génère un nouveau fichier contenant tous les polygones du tableau selon le formalisme précédemment décrit
- 3) Ecrire une fonction qui étant donné un polygone va trier ses points par ordre croissant : la fonction reçoit en paramètre un polygone et retourne un polygone avec les points dans l'ordre. Pour définir la relation d'ordre nous utiliserons l'angle formé par l'axe des abscisses et la droite qui passe par l'origine et un point considéré (angle compris entre 0 et 360°). Un point P2 sera considéré supérieur à P1 si l'angle de P2 est supérieur à l'angle de P1. Pour nous aider nous supposons que nous bénéficions (sans avoir besoin de la définir) d'une fonction *angle* qui, étant donné un point en paramètre, retourne l'angle correspondant. Le choix de l'algorithme de tri est libre.

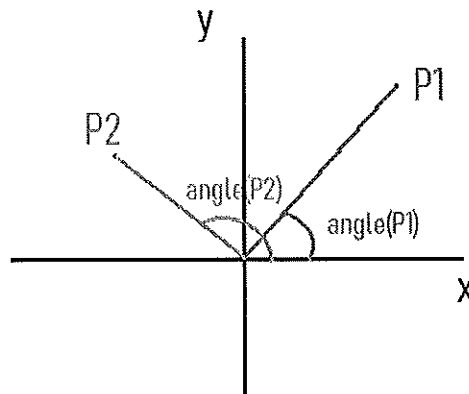


Figure 1 - Illustration de la relation d'ordre entre deux points 2D

Bonus (2pt) : Ecrire la fonction *angle*

¹ Si vous ne parvenez pas à répondre à la question 1, à défaut et pour ne pas être bloqué pour la suite, vous pouvez éventuellement considérer pour les questions 2 et 3 un polygone de N points comme un tableau P de 2xN réels (avec P[0] la coordonnée X du 1^{er} point, P[1] la coordonnée Y du 1^{er} point, P[2] la coordonnée X du 2^{ème} point, P[3] la coordonnée Y du 2^{ème} point, etc.