

I. Points et polygones

Enregistrement *point* :

x, y : réels

nom : caractère

Fin enregistrement

Type *polygone* : tableau de points

Algorithme principal :

Variables :

poly : polygone,

péri: réel,

ℓ_{moy} : réel,

b_{eng} : polygone

poly \leftarrow lecture("poly.txt")

péri \leftarrow calc_péri(poly)

ℓ_{moy} \leftarrow péri / size(poly)

b_{eng} \leftarrow calculer_boîte_englobante(poly)

Afficher péri, ℓ_{moy} , b_{eng}

Procédure lecture(fichier)

Spécification :

Entrée : fichier le nom du fichier à lire

Sortie : le polygone représenté dans le fichier lu

Variables :

poly : réel initialisé à 0,

p_0 : point,

p : point

$p_0 \leftarrow \text{lire_ligne}(\text{fichier})$

poly $\leftarrow [p_0]$

Répéter :

$p \leftarrow \text{lire_ligne}(\text{fichier})$

poly $\leftarrow p :: \text{poly}$

Tant que $p \neq p_0$.

poly $\leftarrow \text{tail}(\text{poly})$

Retourner poly

Procédure calc_péri(poly)

Spécification :

Entrée : poly, un polygone

Sortie : le périmètre du polygone

Variables :

péri : réel initialisé à 0,

x, y, x', y' : réels

$\{x', y'\} \leftarrow \text{poly}[\text{size}(\text{poly}) - 1]$

Pour $\{x, y\} \in \text{poly}$ **faire** :

péri $\leftarrow \text{péri} + \sqrt{(x - x')^2 + (y - y')^2}$

$x' \leftarrow x$

$y' \leftarrow y$

Fin pour

Retourner péri

Procédure calculer_boîte_englobante(poly)

Spécification :

Entrée : poly un polygone

Sortie : la boîte englobante de poly, de type polygone

Variables :

\min_x, \min_y : réels initialisés à $+\infty$

\max_x, \max_y : réels initialisés à $-\infty$

Pour $\{x, y\} \in \text{poly}$ **faire :**

$\min_x \leftarrow \min(x, \min_x)$

$\max_x \leftarrow \max(x, \max_x)$

$\min_y \leftarrow \min(y, \min_y)$

$\max_y \leftarrow \max(y, \max_y)$

Fin pour

Retourner le tableau ci-dessous

$[\{\min_x, \min_y, A\}, \{\max_x, \min_y, B\},$

$\{\max_x, \max_y, C\}, \{\min_x, \max_y, D\}]$

II. Reine et pions

Enregistrement position :

x, y : entiers

Fin enregistrement

Enregistrement échiquier :

pions : tableau de positions

reine : position

Fin enregistrement

Algorithme principal :

Variables :

diag₁, diag₂, diag₃, diag₄, vert₁, vert₂, horiz₁, horiz₂ : huit booléens, initialisés à *F* (false)

reine : position

pions : tableau de positions

x, y, x_r, y_r : entiers

{reine, pions} ← lire_échiquier("chess.txt")

{ x_r, y_r } ← reine

Pour { x, y } ∈ pions faire :

Si $x = x_r$ et $y < y_r$ **alors** horiz₁ ← *V*

Si $x = x_r$ et $y > y_r$ **alors** horiz₂ ← *V*

Si $y = y_r$ et $x < x_r$ **alors** vert₁ ← *V*

Si $y = y_r$ et $x > x_r$ **alors** vert₂ ← *V*

Si $|x - x_r| = |y - y_r|$ **alors**

Si $x > x_r$ et $y < y_r$ **alors** diag₁ ← *V*

Si $x > x_r$ et $y > y_r$ **alors** diag₂ ← *V*

Si $x < x_r$ et $y < y_r$ **alors** diag₃ ← *V*

Si $x < x_r$ et $y > y_r$ **alors** diag₄ ← *V*

Fin si

Fin pour

Afficher

diag₁ + diag₂ + diag₃ + diag₄ + vert₁ + vert₂ + horiz₁ + horiz₂

Procédure lire_échiquier(fichier)

Spécification :

Entrée : fichier, le nom du fichier à ouvrir

Sortie : l'échiquier représenté

Variables :

pions : un tableau de positions

reine : une position

Pour $i \in \llbracket 1, 8 \rrbracket$ faire :

pions[i] \leftarrow lire(fichier)

Fin pour

reine \leftarrow lire(fichier)

Retourner {pions, reine}

Exemple de fichier source (échiquier représenté ci-contre) :

```
1,2 // pion           . . . . .
2,3 // pion           . . . . . P
3,4 // pion           . . R . . P .
3,3 // pion           . . . . .
5,5 // pion           . . . . P . .
6,2 // pion           . P P . . . .
7,6 // pion           P . P . . P .
8,7 // pion           . . . . .
3,6 // reine
```