

T.P. 6

Savez-vous dessiner ?

1 ASCII Art

On veut écrire un programme capable de produire un dessin composé de :

- points
- droites horizontales
- rectangles pleins

On considèrera que l'image est un tableau de L par H pixels. Un pixel est ici un simple caractère qui sera affiché à l'écran. On utilisera par exemple l'espace pour le blanc et le # pour le noir. Tout autre caractère est bien sûr utilisable.

- Quelle structure de données faut-il utiliser pour l'image ? On l'appellera **Image**
- Ecrire une procédure **Point** dont le but est de remplir un pixel donné avec le caractère donné
- Ecrire une procédure **Trait** dont le but est de tracer une droite horizontale d'ordonnée Y, commençant en X1 et finissant en X2. Faire appel à la procédure **Point**
- Ecrire une procédure **Rectangle** qui trace un rectangle de diagonale (X1,Y1)-(X2,Y2) en utilisant la procédure **Trait**
- Faire dans la procédure principale du programme un petit dessin utilisant tous ces types de primitives.

2 Fichiers séquentiels

Plutôt que d'avoir à écrire (et à compiler) un programme pour chaque nouveau dessin, nous allons décrire la scène à dessiner dans un petit langage très simple, qui sera écrit dans un fichier texte. Ce fichier sera lu et analysé par la nouvelle version de notre programme. Si le fichier de description le spécifie, l'image sera elle aussi écrite dans un fichier plutôt que d'être affichée à l'écran.

Le fichier de description est un fichier textuel dont chaque ligne est soit :

- une commande de tracé de point, avec la syntaxe :
P x y c
Où x et y sont des entiers et c, le caractère qui doit être dessiné
- une commande de tracé de rectangle, avec la syntaxe :
R x1 y1 x2 y2 c
Où x1 ... y2 sont des entiers et c, le caractère qui doit être dessiné

- une commande indiquant le nom du fichier de sortie :
S nom-fichier

Chaque ligne commence donc par un caractère qui décrit le type de commande, suivi d'un certain nombre de paramètres séparés par un nombre quelconque de blancs. Les lignes commençant par d'autres caractères que P, R ou S sont simplement ignorées (elles peuvent ainsi être utilisées pour des commentaires). Les lignes vides doivent elles aussi être ignorées.

1.

Ecrire le programme correspondant aux lignes commençant par P et R et pour chaque ligne lue, appeler la procédure correspondante déjà écrite dans l'exercice précédent.

2.

Traiter ensuite le cas de la commande S. Si elle n'existe pas dans le fichier de description, on affiche le dessin à l'écran. Si elle existe, on crée un nouveau fichier et le résultat est produit dans ce fichier.

3.

Si plusieurs commandes S sont présentes dans le fichier, on considère que les commandes R et P entre chaque S s'appliquent à des dessins différents qui seront produits dans des fichiers distincts.