Université BORDEAUX I **Année 1996-97**
Session de septembre

Maîtrise d'informatique

Modèles et Techniques de Programmation (J.P.Braquelaire)

Durée: 2 heures
Tous documents autorisés

Première partie

Le programme suivant est écrit en EMACS LISP. La fonction *defvar* permet de définir un symbole de variable et de lui associer une valeur initiale. La fonction *concat* concatène les éléments qu'elle reçoit en paramètres et produit une chaîne de caractères qui est le résultat de la concaténation. La fonction *format* reçoit une chaîne de description de formatage et une liste de paramètres à formater. Elle produit une chaîne qui est le résultat du formatage. Les spécifications de format s'expriment comme en langage C, par exemple *%s* pour une chaîne, *%d* pour un nombre en notation décimale, *%S* pour un objet LISP quelconque.

f.el

```
(defvar f-debut-liste "(")
(defvar f-format-element "%S")
(defvar f-separateur-elements " ")
(defvar f-fin-liste ")")

(defun f (l)
  (let ((resultat f-debut-liste))
    (if (not (null l))
        (setq resultat
              (concat resultat
                      (format f-format-element (car l)))))
    (while (not (null (cdr l)))
      (setq resultat
            (concat resultat
                    f-separateur-elements
                    (format f-format-element (car (cdr l)))))
      (setq l (cdr l)))
    (concat resultat f-fin-liste)))
```

f.el

1° – **3 points** – Expliquer ce que fait ce programme. Quel est le résultat de l'évaluation de *(f '(a b c d))* ?

2° – **3 points** – Quels sont, selon que le mécanisme de liaison est statique ou dynamique, les deux résultats possibles de l'évaluation de l'expression

```
(let ((f-debut-liste "{")
      (f-format-element "\"%S\"")
      (f-separateur-elements ", ")
      (f-fin-liste " }"))
  (f '(a b c d)))
```

Expliquer pourquoi. Quel est ce résultat sous EMACS LISP ?

3° **– 2 points –** Commenter, en s'appuyant sur cet exemple, l'intérêt du mécanisme de liaison dynamique du point de vue de la réutilisabilité. Quel risque d'erreur cela peut-il présenter et comment le limiter ?

Deuxième partie

4° **– 3 points –** On considère l'allocation d'une matrice rectangulaire *m* de taille $\Delta X \times \Delta Y$ éléments de type *t*. Comment effectue-t-on cette allocation ? Comment accède-t-on aux éléments de la matrice ?

5° **– 2 points –** On suppose que la matrice *m* code une image et que le type *t* est le type d'un pixel. La fonction standard *fwrite* permet d'écrire le contenu d'une zone mémoire sur un flot d'entrées/sorties. Son synopsys est :

```
size_t fwrite(const void *tampon, size_t taille, size_t nombre, FILE *flot)
```

tampon étant l'adresse de la zone mémoire à écrire, ***nombre*** et ***taille*** le nombre d'éléments à écrire et la taille d'un élément, et ***flot*** le flot d'entrées/sorties. Comment faut-il organiser en mémoire les lignes et les colonnes de la matrice pour que l'appel de fonction

fwrite(m, sizeof(t), $\Delta X \times \Delta Y$*, flot)*

écrive l'image de telle sorte que les pixels soient ordonnés ligne par ligne ?

6° **– 7 points –** Une image PGM est une image monochrome dont chaque pixel est un entier codant un niveau de gris. Donner l'implémentation du module *pgm* comportant les fonctions d'interface suivantes :

- *pgm pgm_creer(size_t largeur, size_t hauteur)* et *void pgm_liberer(pgm image)* : allocation et libération d'un objet *pgm* permettant de gérer une image.
- *pixel pgm_lire_pixel(pgm image, size_t i, size_t j)* : lecture de la valeur du pixel de coordonnées (i, j).
- *void pgm_ecrire_pixel(pgm image, size_t i, size_t j, pixel p)* : modification de la valeur du pixel de coordonnées (i, j).
- *void pgm_ecrire_image(pgm image, FILE *flot)* : écriture sur le flot d'un fichier image au format PGM, selon le format suivant :

  P5
  largeur hauteur
  nombre-de-niveaux-de-gris
  $p_{1,1}p_{2,1}\cdots p_{\Delta X,1}, p_{1,2}, p_{2,2}, p_{3,2}, \ldots, p_{\Delta X,\Delta Y}$

largeur, ***hauteur*** et ***nombre-de-niveaux-de-gris*** étant des chaînes de caractères numériques et le $p_{i,j}$ des valeurs binaires écrites sur un octet sans séparateur (dans ce format, la valeur maximale de ***nombre-de-niveaux-de-gris*** est 255).