

LIFAP1 – TP6 : Les structures

Objectifs : Manipulation des structures

Petit exercice d'échauffement

Une fleur est définie par :

- ✓ son nom,
- ✓ sa couleur,
- ✓ **un tableau** comparatif de son prix chez 3 commerçants sélectionnés.

Une liste de fleurs est définie par :

- ✓ le nombre de fleurs qu'elle contient,
- ✓ un tableau de fleurs.

1. Définir en langage C/C++ une constante CHMAX ayant pour valeur 50 qui sera utilisée comme taille maximale des chaînes de caractères.
2. Définir en langage C/C++ les structures de données `fleur` et `liste_fleurs` permettant de stocker toutes les informations concernant une fleur et une liste de fleurs.
3. Écrire en langage C/C++ un sous-programme `saisir_fleur` permettant de saisir toutes les informations relatives à une fleur. Attention on recommencera la saisie tant que les données de prix ne sont pas strictement positives.
4. Écrire en langage C/C++ un sous-programme `saisir_liste_fleurs` permettant de saisir toutes les informations relatives à une liste de fleurs. On utilisera pour cela le sous-programme de la question 3.
5. Afin d'effectuer des comparaisons tarifaires entre les 3 commerçants sélectionnés, on souhaite connaître pour une liste de fleurs, le prix total de toutes les fleurs chez chacun des 3 commerçants. Écrire en langage C/C++ **un seul** sous-programme `prix_liste_fleurs` permettant d'extraire ces 3 informations. On s'assurera de ne parcourir qu'une seule fois le tableau de fleurs.
6. Écrire en langage C/C++ un sous-programme `affiche_fleurs_rouges` permettant d'afficher le nom de toutes les fleurs de couleur rouge de la liste de fleurs.
7. Écrire en langage C/C++ le programme principal permettant de remplir un tableau avec les caractéristiques de 5 fleurs, d'afficher le nom de toutes les fleurs rouges de cette liste et d'afficher le numéro du commerçant le moins cher des 3 pour cette liste.

Des images en mode texte

1. Représentation des images à l'aide d'une structure.
 - a. Définissez en langage C la structure *Image* représentant une image de caractères.
Voici sa définition en langage algorithmique :

```
Constantes : DIMMAX : Entier = 100
Structure Image
    dimx, dimy : Entier // les dimensions de l'image <= DIMMAX
    im : tableau[DIMMAX][DIMMAX] de caractères
Fin Structure
```
 - b. Ecrivez une procédure permettant d'initialiser une image *im* avec le caractère *c*.
 - c. Écrivez la procédure *imAff* qui efface l'écran et affiche l'image.
Attention : il faut afficher l'image de haut en bas car l'écran se remplit de haut en bas!
Rappel : pour effacer l'écran l'instruction est `system("cls");`

2. Dessins (boucle avec bornes évoluées).

- a. Écrivez la procédure *imDessineRect* qui remplit un rectangle avec le caractère *c* dans l'image. Cette procédure n'affiche pas l'image !

// En algorithmique

Procédure *imDessineRect*(*im* : donnée-résultat Image; *c* : donnée Caractère; *xmin*, *ymin*, *xmax*, *ymax* : donnée Entier)

Remarque : L'image n'est pas initialisée entre chaque dessin. Deux figures tracées à la suite dans une image vont se superposer.

- b. Écrivez la procédure *imDessineCercle* qui dessine un cercle plein dans l'image.

// En algorithmique

Procédure *imDessineCercle*(*im* : donnée-résultat Image; *c* : donnée Caractère; *x*, *y*, *r* : donnée Entier)

Version simple de l'algorithme : parcourez toutes les cases de l'image, si la case considérée a une distance au centre du cercle inférieure au rayon, la case est dans le cercle.