

Allocation dynamique Travaux Pratiques – Séance n° 12

1) Soient le type `forme` et la procédure `afficherForme` suivants :

```
typedef enum { carre, rond, triangle } forme;

void afficherForme(forme f) {
    switch (f) {
        case carre : printf("carré"); break;
        case rond : printf("rond"); break;
        case triangle : printf("triangle"); break;
        default : printf("forme inconnue");
    }
}
```

Vous testerez les exercices suivants au fur et à mesure.

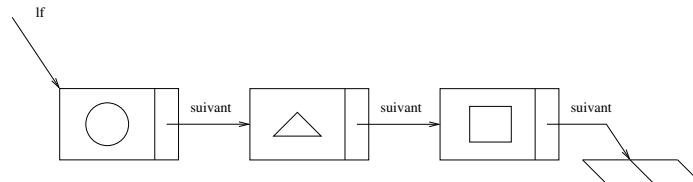
- Déclarez trois pointeurs, `p`, `q` et `r` sur le type `forme`.
- Allouez dynamiquement deux objets, un carré et un rond, accessibles respectivement par les pointeurs `p` et `q`.
- Affichez la valeur des objets pointés par `p` et `q`.
- Faites pointer `r` sur l'objet pointé par `p`, et affichez la valeur de l'objet pointé par `r`.
- Transformez la valeur de l'objet pointé par `p` en `triangle` et affichez la valeur de l'objet pointé par `r`.
- Détruisez l'objet pointé par `p`.
- Affichez la valeur de l'objet pointé par `r`. Expliquez le résultat.

2) Soient les déclarations suivantes :

```
typedef struct noeud {
    forme f;
    struct noeud *suivant;
} * ListeDeFormes;
```

ListeDeFormes lf;

- Écrivez dans la fonction `main` les instructions nécessaires pour créer la structure suivante :



- À l'aide de la procédure `afficherForme`, écrivez une procédure `afficherListeDeFormes` qui affiche la liste des formes. Cette procédure possède un seul paramètre de type `ListeDeFormes`.
- Affichez le contenu de `lf`.
- Ajoutez en tête de `lf` un `carre` et affichez à nouveau `lf`.
- Supprimez de la liste l'objet dynamique qui contient le triangle et affichez à nouveau `lf`.

3) Écrivez la fonction `longueur` qui renvoie le nombre de formes d'une liste de formes.

```
/* Rôle : renvoie le nombre de forme de la liste */
int longueur(ListeDeFormes l)
```

4) Écrivez la fonction `ième` qui renvoie la forme à un rang donné d'une liste de formes.

```
/* Rôle : renvoie la forme de rang r de la liste l */
/* Antécédent : 1 ≤ r ≤ longueur(l) */
forme ième(ListeDeFormes l, int r)
```

5) Écrivez la fonction `insérer` qui insère une forme à un rang donné d'une liste de formes.

```
/* Rôle : insérer dans la liste l la forme f, au rang r */
/* Antécédent : 1 ≤ r ≤ longueur(l) + 1 */
void insérer(ListeDeFormes *l, int r, forme f)
```

6) Écrivez la fonction `supprimer` qui supprime la forme d'un rang donné d'une liste de formes.

```
/* Rôle : supprime la forme de rang r de la liste l */
/* Antécédent : 1 ≤ r ≤ longueur(l) */
void supprimer(ListeDeFormes *l, int r)
```

7) Refaites l'exercice 2 avec les fonctions `insérer` et `supprimer` que vous venez d'écrire.