

```
typedef struct point {  
    double x, y;  
} Point2D;
```

```
typedef enum { Cercle, Rectangle } TypeFigure;
```

```
typedef struct {  
    TypeFigure type;           /* qui est quoi ! */  
    Point2D origine;          /* commun a toutes les figures */  
    double rayon;             /* specifique aux cercles */  
    double largeur, longueur; /* specifique aux rectangles */  
} Figure;
```

Pour `origine`, on pourrait avoir un pointeur sur le point plutôt que le point lui-même selon ce qu'on veut faire.

```
void imprime(Figure fig) {  
    switch (fig.type) {  
        case Cercle:  
            printf("Cercle de rayon : %g\n", fig.rayon);  
            break;  
        case Rectangle:  
            printf("Rectangle de largeur: %g et longueur : %g\n",  
                fig.largeur, fig.longueur);  
            break;  
        default: fprintf(stderr, "Figure inconnue !\n");  
    }  
}
```

```
double perimetre(Figure fig) {  
    switch(fig.type) {  
        case Cercle:      return (2*3.14*fig.rayon);  
        case Rectangle:  return (2*(fig.largeur+fig.longueur));  
        default: fprintf(stderr, "Figure inconnue !\n");  
                    return -1;  
    }  
}
```

Version améliorée (Point2D et TypeFigure inchangés)

```
typedef struct attC { /* attributs des cercles */  
    double rayon;  
} AttCercle;
```

```
typedef struct attR { /* attributs des rectangles */  
    double largeur, longueur;  
} AttRectangle;
```

```
typedef union { /* choix entre les possibilités */  
    AttRectangle rect;  
    AttCercle cercle;  
} AttSpecifiques;
```

```
typedef struct figures {  
    TypeFigure type;  
    Point2D origine; /* commun a toutes les figures */  
    AttSpecifiques att; /* la partie specifique */  
} Figure;
```

```
double perimetre(Figure fig) {  
    switch (fig.type) {  
        case Cercle:    return (2*3.14*fig.att.cercle.rayon);  
        case Rectangle: return (2*(fig.att.rect.largeur +  
                                fig.att.rect.longueur));  
        default: printf("Figure inconnue !");  
                return -1;  
    }  
}
```