Auteur : Mohamed Messabihi
Matière : Programmation et structures de données
Date 18 Mai 2017
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

Affichage — 3 pts. 15'

Qu'affiche le programme suivant ?

```c
#include <stdio.h>
void main()
{
    char s[100] = "TotoLoulou";
    while ( strchr(s, 'o') != NULL ) strcpy(strchr(s, 'o'), strchr(s, 'o')+1 );
    printf("s = %s",s);
}
```

Allers-retours — 6 pts. 25'

Écrire une fonction qui transforme un tableau TAB de deux dimensions (N lignes et M colonnes) en un tableau linéaire A (à une seule dimension L=N*M). La fonction affecte les valeurs dans le tableau linéaire A en parcourant en allers-retours le tableau à deux dimensions TAB. Autrement dit, la fonction parcourra la première ligne de TAB de gauche à droite puis la seconde de droite à gauche, la troisième de gauche à droite et ainsi de suite en alternant, à chaque fois, le sens de parcours des lignes.

Exemple :

TAB

11	2	8	4
19	7	13	9
6	14	10	3
16	1	12	5
18	17	20	15

↓

A

11	2	8	4	9	13	7	19	6	14	10	3	5	12	1	16	18	17	20	15

Rectangles — 11 pts. 50'

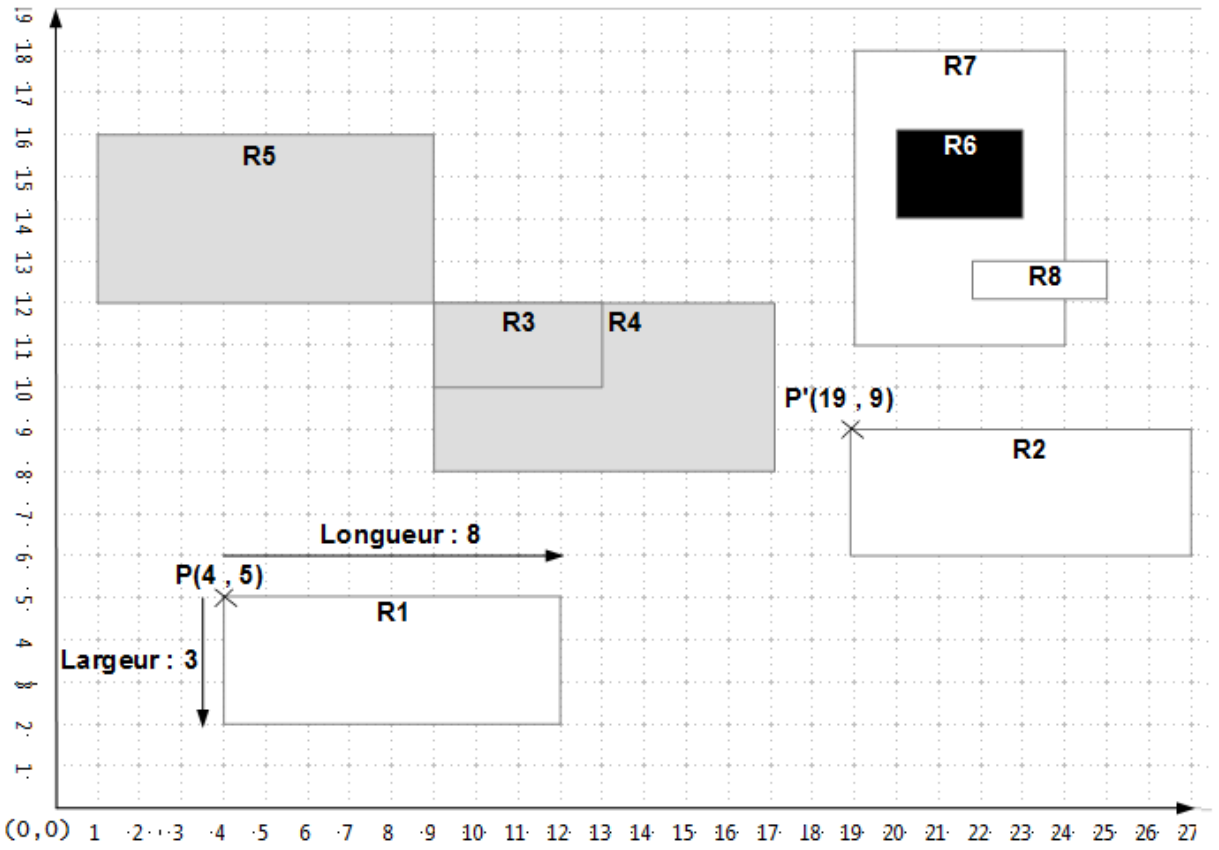
On souhaite programmer un éditeur graphique qui permet de dessiner des rectangles sur un plan muni d'un repère cartésien.

Un rectangle est défini par :

— un point sur le plan qui représente son coin supérieur gauche ;
— une longueur ;
— une largeur ;
— et une couleur qui peut être blanche, noire ou grise.

Un point P est défini par un couple de réels :
— x appelé l'abscisse de P ;
— y appelé l'ordonnée de P.

1. Définir la structure Rectangle. **3 pts**
2. Écrire une fonction saisir_Rectangle. **2 pts**
3. Écrire une fonction deplacer qui prend en entrée un rectangle R et un point P′ et puis elle déplace le rectangle R vers le point P′. Par exemple, le rectangle R1, de la figure ci-dessous, prendra la place du rectangle R2 s'il est déplacé au point P′. **1 pt**
4. Écrire une fonction zoomer qui prend en entrée un rectangle et un coefficient de zoom. La fonction maintient le coin supérieur gauche du rectangle puis elle agrandie (ou elle rapetisse) les dimensions du rectangle en fonction du coefficient de zoom. Par exemple, le rectangle R3 devient égale au rectangle R4 s'il est zoomé avec un coefficient égale à 2 et inversement R4 devient égale à R3 s'il est (dé)zoomé avec un coefficient égale à 0.5. **1 pt**
5. Écrire une fonction symétrique qui prend en entrée un rectangle R et retourne le rectangle symétrique R′ du R par rapport à son coin supérieur gauche. Par exemple, le rectangle R5 est le résultat de l'appel de cette fonction avec le rectangle R4 comme paramètre. **1 pt**
6. Écrire une fonction inclut qui prend en entrée deux rectangles R1 et R2 et renvoie 1 si R2 est totalement inclus dans R1, 0 sinon. Par exemple R6 est totalement inclus dans R7 mais R8 ne l'est pas. **2 pts**



7. Écrire une fonction **main** qui permet de : **1pt**
   — déclarer un Graph qui contient un ensemble de rectangles.
   — saisir le rectangle R1 de la figure ci-dessus dans le premier du Graph,
   — déplacer le premier rectangle au point P′(19, 9),
   — et enfin, ajouter un deuxième rectangle (dans Graph) symétrique au premier rectangle déplacé.

« Bon courage »

Auteur : Mohamed Messabihi
Matière : Programmation et structures de données
Date 18 Mai 2017
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

Affichage — 3 pts. 15'

Qu'affiche le programme suivant ?

```c
#include <stdio.h>
void main()
{
    char s[100] = "TotoLoulou";
    while ( strchr(s, 'o') != NULL ) strcpy(strchr(s, 'o'), strchr(s, 'o')+1 );
    printf("s = %s",s);
}
```

Solution

Affichage

```
s = TtLulu
```

Allers-retours — 6 pts. 25'

Écrire une fonction qui transforme un tableau TAB de deux dimension (N lignes et M colonnes) en un tableau linéaire A (à une seule dimension L = N*M). La fonction affecte les valeurs dans le tableau linéaire A en parcourant en allers-retours le tableau à deux dimensions TAB. Autrement dit, la fonction parcourra la première ligne de TAB de gauche à droite puis la seconde de droite à gauche, la troisième de gauche à droite et ainsi de suite en alternant, à chaque fois, le sens de parcours des lignes.

Exemple :

TAB

11	2	8	4
19	7	13	9
6	14	10	3
16	1	12	5
18	17	20	15

↓

A

11	2	8	4	9	13	7	19	6	14	10	3	5	12	1	16	18	17	20	15

Solution

```c
void Allers_retours(int TAB[10][10], int A[100] , int n, int m)
{
  int i, j, k=0 ;
  for (i=0 ; i<n ; i++)
   {
     if (i%2 == 0)
            for (j=0 ; j<m ; j++) A[k++] = TAB[i][j];
     else
            for (j=m-1 ; j>=0 ; j--) A[k++] = TAB[i][j];
   }
}
```

Rectangles — 11 pts. 50'

On souhaite programmer un éditeur graphique qui permet de dessiner des rectangles sur un plan muni d'un repère cartésien.

Un rectangle est défini par :
- un point sur le plan qui représente son coin supérieur gauche ;
- une longueur ;
- une largeur ;
- et une couleur qui peut être blanche, noire ou grise.

Un point P est défini par un couple de réels :
- x appelé l'abscisse de P ;
- y appelé l'ordonnée de P.

1. Définir la structure Rectangle. **3 pts**

Solution

```c
typedef enum Couleur Couleur ;
enum Couleur {BLANC, NOIR, GRIS};

typedef struct Point Point ;
struct Point {
    double x;
    double y;
};
typedef struct Rectangle Rectangle ;
struct Rectangle {
    Point coin ;
    int longueur;
    int largeur;
    Couleur couleur;
};
```

2. Écrire une fonction saisir_Rectangle. **2 pts**

Solution

```c
void saisir_Point(Point *p){
    puts("Donnez le x : ");
    scanf("%lf", &p->x);
    puts("Donnez le y : ");
    scanf("%lf", &p->y);
}
```

```c
void saisir_Rectangle(Rectangle *r){
    puts("Donnez les coordonnees du coin superieur gauche : ");
    saisir_Point(&r->coin);
    puts("Donnez la longueur : ");
    scanf("%d", &r->longueur);
    puts("Donnez la largeur : ");
    scanf("%d", &r->largeur);
    puts("Donnez la couleur, 0 : Blanc, 1 : Noir, 2 : Gris");
     scanf("%d", &r->couleur);
}
```

3. Écrire une fonction deplacer qui prend en entrée un rectangle R et un point P/ et puis elle déplace le rectangle R vers le point P/. Par exemple, le rectangle R1 prendra la place du rectangle R2 s'il est déplacé au point P'. **1 pt**

Solution

```c
void deplacer(Rectangle *r, Point p)
{
    r->coin = p;
}
```

4. Écrire une fonction zoomer qui prend en entrée un rectangle et un coefficient de zoom. La fonction maintient le coin supérieur gauche du rectangle puis elle agrandie ou elle rapetisse les dimensions du rectangle en fonction du coefficient du zoom. Par exemple, le rectangle R3 devient égale au rectangle R4 s'il est zoomé avec un coefficient égale à 2 et inversement R4 devient égale à R3 s'est zoomé avec un coefficient égale à 0.5. **1 pt**

Solution

```c
void zoomer(Rectangle *r, int zoom)
{
    r->longueur *= zoom;
    r->largeur *= zoom;
}
```

5. Écrire une fonction symétrique qui prend en entrée un rectangle R et retourne le rectangle symétrique R/ du R par par rapport son coin supérieur gauche. Par exemple, le rectangle R5 est le résultat de l'appel de cette fonction avec le rectangle R4 comme paramètre. **1 pt**

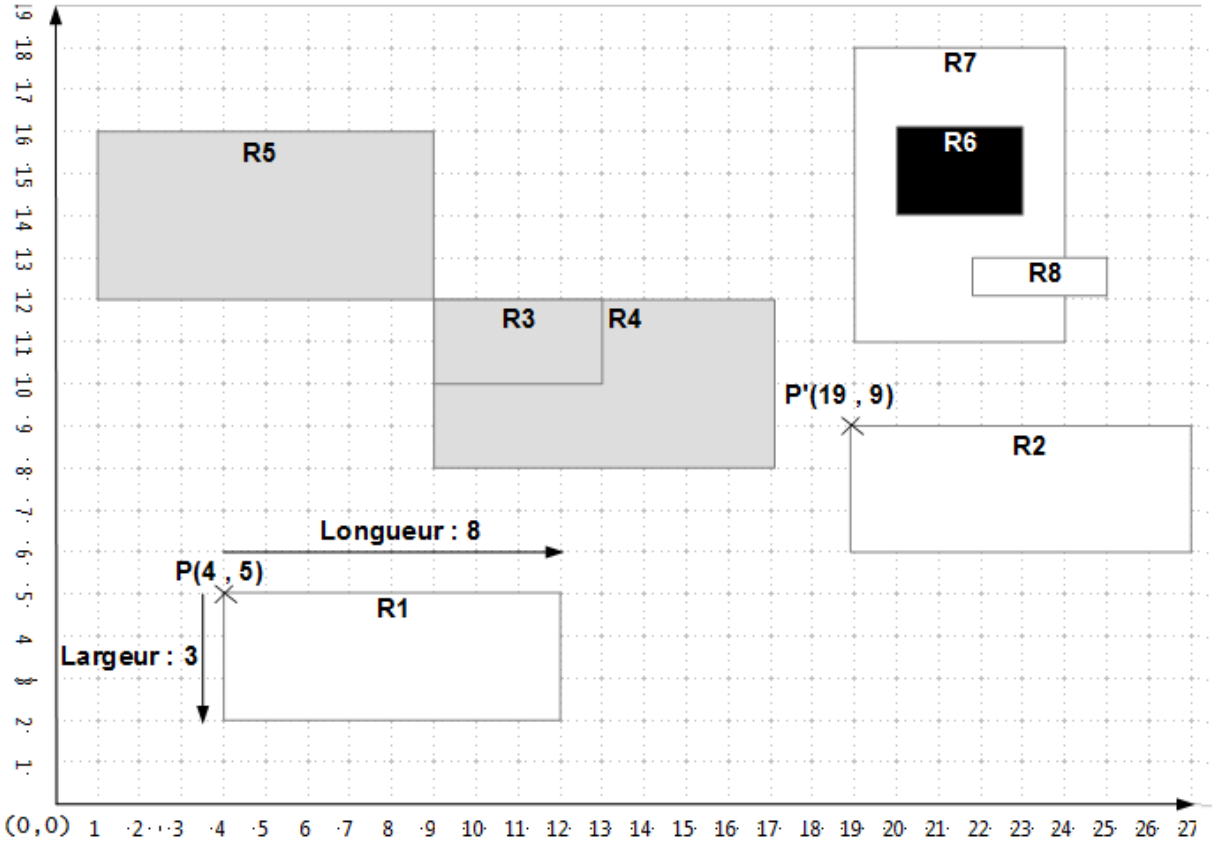
Solution

```c
Rectangle symetrique(Rectangle r)
{
    Rectangle sym_r;
    sym_r.coin.x = r.coin.x - r.longueur;
    sym_r.coin.y = r.coin.y + r.largeur;
    sym_r.longueur = r.longueur;
    sym_r.largeur = r.largeur;
    sym_r.couleur = r.couleur;
    return sym_r;
}
```

6. Écrire une fonction inclut qui prend en entrée deux rectangles R1 et R2 et renvoie 1 si R2 est inclus dans R1, 0 sinon. Par exemple R6 est inclus dans R7 mais R8 ne l'est pas. **2 pts**

Solution

```c
int inclut(Rectangle r1, Rectangle r2 )
{
    if((r2.coin.x >= r1.coin.x) && (r2.coin.y <= r1.coin.y) &&
       (r2.coin.x+r2.longueur <= r1.coin.x+r1.longueur) &&
       (r2.coin.y-r2.largeur >= r1.coin.y-r1.largeur))
        return 1;
    else return 0;
}
```



7. Écrire une fonction **main** qui permet de : **1pt**
   - déclarer un Graph qui contient un ensemble de rectangles.
   - saisir le rectangle R1 de la figure ci-dessus dans le premier rectangle de la variable Graph,
   - déplacer le premier rectangle au point P′(19, 9),
   - et enfin, ajouter un deuxième rectangle (dans Graph) symétrique au premier rectangle déplacé.

Solution

```c
void main(){
    Rectangle Graph[10];
    Point P = {19,9};
    saisir_Rectangle(&Graph[0]);
    deplacer(&Graph[0], P);
    Graph[1] = symetrique(Graph[0]);
}
```

« Bon courage »