

Examen – Session de rattrapage

Matière : Atelier programmation II
 Enseignants : Majdi JRIBI et Dorsaf SEBAI
 Filière : MPI
 Nombre de pages : 2 pages

Semestre: Second semestre
 Date: 15 Juin 2017
 Durée: 1h30
 Documents : non autorisés

Chaque exercice doit être obligatoirement rédigé sur une feuille d'examen à part

Exercice 1 : Arbre binaire et arbre binaire de recherche

Considérons les déclarations suivantes :

```
typedef struct arb
```

```
{    int val;
    struct arb *fg;
    struct arb *fd;
} arb;
```

```
typedef arb* arbre;
```

1- Ecrire en langage C une fonction récursive *void detruire_arbre (arbre ar)* qui prend en entrée un arbre binaire et qui permet de libérer la mémoire occupée par tous les nœuds de cet arbre.

2- Ecrire en langage C une fonction *int est_binaire_recherche(arbre ar₁)* qui prend en entrée un arbre binaire et qui renvoie la valeur 1 si l'arbre ar₁ est binaire de recherche sinon elle renvoie la valeur 0.

Exercice 2 : Arbre binaire

Considérons les déclarations suivantes d'un arbre binaire:

```
typedef struct arb
```

```
{    int val;
    struct arb *fg;
    struct arb *fd;
} arb;
```

```
typedef arb* arbre;
```


Ecrire en langage C une fonction itérative qui cherche un élément dans un arbre binaire quelconque. Cette fonction retourne 1 si l'élément est trouvé et 0 sinon.

Exercice 3 : Arbre binaire

Considérons les déclarations suivantes d'un arbre binaire:

```
typedef struct arb
```

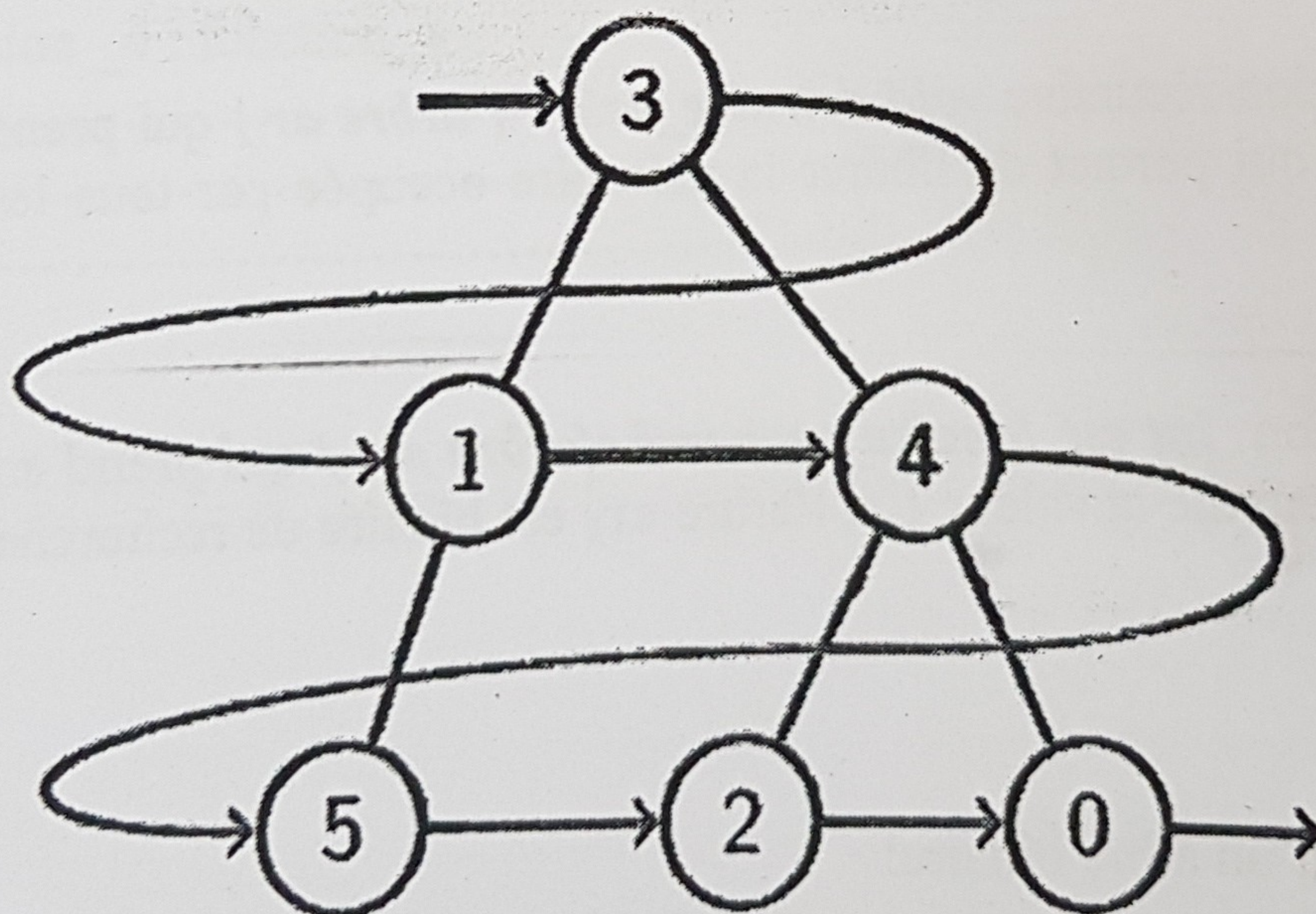
```
{    int val;  
    struct arb *fg;  
    struct arb *fd;  
} arb;
```

```
typedef arb* arbre;
```

Il existe deux types de parcours pour un arbre binaire : Un parcours en profondeur et un parcours en largeur. Nous avons déjà vu en cours le parcours en profondeur qui lui-même se décompose en trois sous parcours : préfixé, infixé et postfixé. Le parcours en largeur est réalisé quant à lui de la manière suivante : on commence par explorer la racine, puis ses fils, puis les fils des fils, etc.

Remarque : l'exploration des fils d'un nœud se fait de gauche à droite

Exemple de parcours en largeur d'un arbre binaire :



Si lors de ce parcours en largeur on affiche les valeurs des nœuds, le résultat obtenu est le suivant :

3, 1, 4, 5, 2, 0

Ecrire une fonction itérative en langage C qui permet de réaliser le parcours en largeur d'un arbre binaire et afficher les valeurs de ses nœuds suite à ce parcours.