

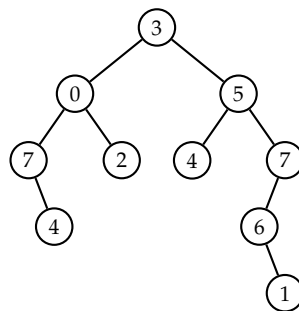
## Recorreguts recursius d'un arbre

P57669\_ca

Feu un programa que llegeixi la descripció d'un arbre binari de naturals i escrigui els seus recorreguts en postordre i inordre.

Tant en aquest exercici com en la resta d'exercicis d'aquesta secció, si no es diu el contrari la descripció d'un arbre consisteix en el nombre de nodes  $n$  seguit del recorregut en preordre, el qual inclou les fulles marcades amb un -1. Aquest recorregut té doncs  $2n + 1$  elements.

Per exemple, l'arbre



es descriuria amb 10 3 0 7 -1 4 -1 -1 2 -1 -1 5 4 -1 -1 7 6 -1 1 -1 -1 -1

Per resoldre tant aquest exercici com la majoria d'aquesta secció, us caldrà emmagatzemar l'arbre en un vector. Feu-ho usant aquest codi (lleugerament modificat si cal):

```

struct Node {
    int valor;
    int esq, dre;
};

// Llegeix un arbre i el desa a un tros del vector v començant a la posició j.
// Modifica la variable j perquè apunti a la següent posició lliure de v.
// Retorna la posició dins de v de l'arrel del (sub)arbre llegit (o -1).
int arbre(int& j, vector<Node>& v) {
    int x;
    cin >> x;
    if (x == -1) return -1;

    int a = j;
    ++j;
    v[a].valor = x;
    v[a].esq = arbre(j, v);
    v[a].dre = arbre(j, v);
    return a;
}

...

int main() {

```

```

    int n;
    cin >> n;
    vector<Node> v(n);
    int j = 0;
    int a = arbre(j, v);
    ...
}

```

Amb l'arbre de l'exemple, el contingut final del vector  $v$  seria

posició	0	1	2	3	4	5	6	7	8	9
$v.valor$	3	0	7	4	2	5	4	7	6	1
$v.esq$	1	2	-1	-1	-1	6	-1	8	-1	-1
$v.dre$	5	4	3	-1	-1	7	-1	-1	9	-1

Fixeu-vos que cada posició del vector guarda el valor d'un node, així com la posició del seu fill esquerre i del seu fill dret. El valor -1 s'usa per indicar els arbres buits. La variable  $a$  del programa principal és la posició de l'arrel de l'arbre, i val -1 si l'arbre és buit, i 0 si no ho és.

## Entrada

L'entrada consisteix en la descripció d'un arbre binari de naturals.

## Sortida

Escriviu dues línies, amb els recorreguts en postordre i inordre de l'arbre. Cada element ha de sortir precedit d'un espai.

### Exemple d'entrada

```

10
3 0 7 -1 4 -1 -1 2 -1 -1 5
4 -1 -1 7 6 -1 1 -1 -1 -1

```

### Exemple de sortida

```

4 7 2 0 4 1 6 7 5 3
7 4 0 2 3 4 5 6 1 7

```

## Informació del problema

Autor : Salvador Roura  
Generació : 2024-05-02 20:00:52

© [Jutge.org](https://jutge.org), 2006–2024.  
<https://jutge.org>