

STRUTTURE DATI e LABORATORIO II

Esercitazione n° 14

Albero di ricerca binario

Scrivere un algoritmo C che implementi la struttura *albero di ricerca binario*.

Il programma deve prevedere le funzioni di inserimento, cancellazione, visualizzazione e ricerca.

Il campo chiave è costituito da una stringa.

Suggerimento. Per semplicità si supponga che ogni dato sia una stringa che ne costituisce la chiave.

Per visualizzare i dati utilizzare gli algoritmi che visitano l'albero. Implementare inoltre sia l'algoritmo di ricerca ricorsivo che quello iterativo.

L'algoritmo può essere strutturato con un menù del tipo

```
int menu(void)
{
    printf("\n\nIndica l'operazione da eseguire:\n\n");
    printf("inserisci elemento          --> 1\n");
    printf("ricerca elemento (ric.)           --> 2\n");
    printf("ricerca elemento (iter.)          --> 3\n");
    printf("cancella elemento per fusione      --> 4\n");
    printf("cancella elemento per copiatura    --> 5\n");
    printf("visita inorder                     --> 6\n");
    printf("visita preorder                    --> 7\n");
    printf("visita postorder                   --> 8\n");
    printf("termina                            --> 9\n");
    scanf("%d",&buf);
    fflush();
    return buf;
}
```

IMPORTANTE. Si può implementare anche un solo algoritmo di cancellazione!!

Per la cancellazione per copiatura si può utilizzare la funzione seguente:

```

void delete_nodo2(tree_pointer *root, char *item)
// Cancellazione per copiatura
{
    tree_pointer tmp, node, prev=NULL, p=*root, previous;

    while (p!=NULL && strcmp(item, p->dati)!=0)
    {
        prev = p;
        if(strcmp(item, p->dati) > 0 )    p = p->destro;
        else    p = p->sinistro;
    }
    node = p;
    if ( p!=NULL && strcmp(item, p->dati)==0) {
        if (node->destro==NULL)    node = node->sinistro;
        else if (node->sinistro==NULL)    node=node->destro;
        else {
            tmp = node->sinistro;
            previous = node;
            while (tmp->destro!=NULL)
            {
                previous = tmp;
                tmp = tmp->destro;
            }
            strcpy(node->dati, tmp->dati);
            if (previous==node) previous->sinistro = tmp->sinistro;
            else previous->destro = tmp->sinistro;
        }
        if (p==*root)    *root = node;
        else if (prev->sinistro == p)    prev->sinistro = node;
        else    prev->destro = node;
    }
    else if (*root != NULL) printf("\nStringa non trovata!!");
    else    printf("\nL'albero e' vuoto!!");
return;
}

```

Buon lavoro!