```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

struct node{
   int data;
   struct node *right;
   struct node *left;
};

struct node* search(struct node *root, int x)
{
   if(root==NULL || root->data==x)
      return root;
   else if(x < root->data)
      return search(root->right, x);
   else
      return search(root->left,x);
}

struct node* find_minimum(struct node *root)
{
   if(root == NULL)
      return NULL;
   else if(root->left != NULL)
      return find_minimum(root->left);
   return root;
```

```
}

struct node* new_node(int x)
{
   struct node *p;
   p = malloc(sizeof(struct node));
   p->data = x;
   p->left = NULL;
   p->right = NULL;

   return p;
}

struct node* insert(struct node *root, int x)
{
   if(root==NULL)
      return new_node(x);
   else if(x < root->data)
      root->right = insert(root->right, x);
   else
      root->left = insert(root->left,x);
   return root;
}

struct node* delete(struct node *root, int x)
{
   if(root==NULL)
```

```
    return NULL;
if (x < root->data)
    root->right = delete(root->right, x);
else if(x > root->data)
    root->left = delete(root->left, x);
else{
    if(root->left==NULL && root->right==NULL){
        free(root);
        return NULL;
    }
    else if(root->left==NULL || root->right==NULL){
        struct node *temp;
        if(root->left==NULL)
            temp = root->right;
        else
            temp = root->left;
        free(root);
        return temp;
    }
    else
    {
        struct node *temp = find_minimum(root->right);
        root->data = temp->data;
        root->right = delete(root->right, temp->data);
    }
}
return root;
```

```
}

void sort(struct node *root)
{
   if(root!=NULL){
      sort(root->right);
      printf("%d ", root->data);
      sort(root->left);
   }
}

void printArray(int array[],int size){
        printf("Numbers : ");
        int i;
        for(i=0; i<size; i++){
                printf("%d ",array[i]);
        }
        printf("\n");
}

void printTree( struct node* root, int spaces )
{

  int i;
  if( root != NULL )
  {
    printTree( root->left, spaces + 3 );
```

```
   for( i = 0; i < spaces; i++ )
                        printf(" ");
   printf("%d\n",root->data );
   printTree( root->right, spaces + 3 );
  }

}

struct node *add(struct node *root){
        int i, array[10];
        for(i=0; i<10; i++){
                array[i] = rand()%100+1;
        }
        printArray(array,10);
        root = new_node(array[0]);

        for(i=1; i<10; i++){
                insert(root,array[i]);
        }
        printf("\n---------------------------------------\n");
        printTree(root,0);
        printf("\n---------------------------------------\n");
        return root;
}

int main()
```

```
{
        srand(time(0));
        struct node *root = NULL;

        root = add(root);
        printf("\n");

        printf("Sort\t: ");
        sort(root);

        printf("\n");

    return 0;
}
```