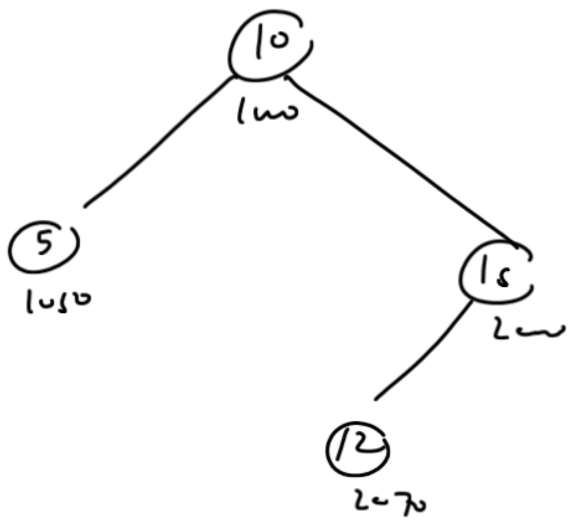


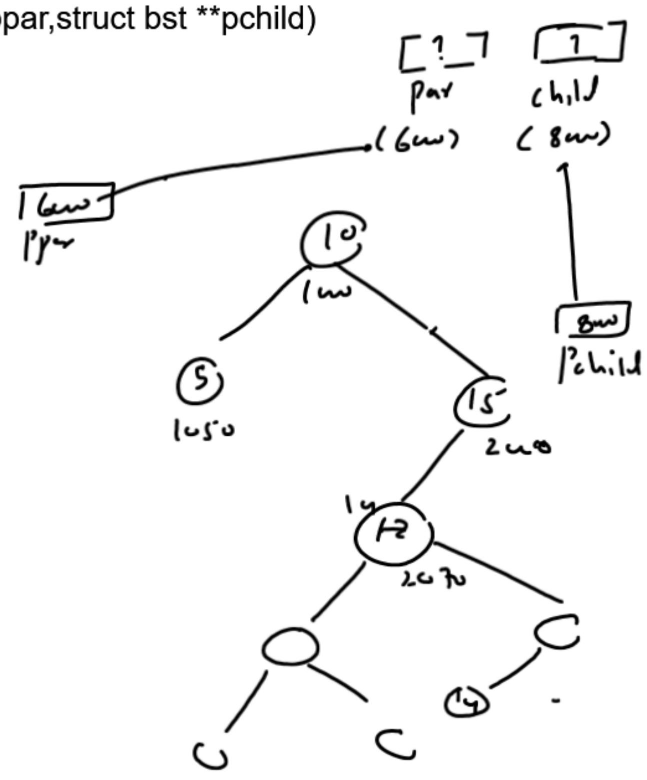
```
struct bst
{
   struct bst *left;
   int data;
   struct bst *right;
};
struct Stack
{
   struct bst* arr[10];
   int tos;
};
void push(struct Stack*,struct bst*);
struct bst* pop(struct Stack*);
void append(struct bst **,int);
void inorder(struct bst*);
```
int search(struct bst*,int,struct bst,struct bst**);**
void del(struct bst,int);**

```
int main()
{
   struct bst *root=NULL;
   append(&root,10);
   append(&root,7);
   append(&root,12);
   append(&root,5);
   ....
   inorder(root);

   del(&root,12);
   inorder(root);
   return 0;
}
```

```
int search(struct bst*p,int x,struct bst**ppar,struct bst **pchild)
{
   struct bst *q=NULL;
   while(p!=NULL)
   {
       if(p->data==x)
         {
              *pchild=p;
              *ppar=q;
              return 1;
        }
      q=p;
      if(p->data>x)
           p=p->left;
     else
          p=p->right;
   }
return 0;
}
```

14



```
void del(struct bst **pr,int x)
{
   struct bst *par,*child,*q;
   int ans;
   if(*pr==NULL)
     {
        printf("Empty tree");
        return;
     }
   ans=search(*pr,x,&par,&child);
   if(ans==0)
      {
          printf("Node not found!");
          return;
     }
if(child->left!=NULL && child->right!=NULL)
{
    // logic
}
```