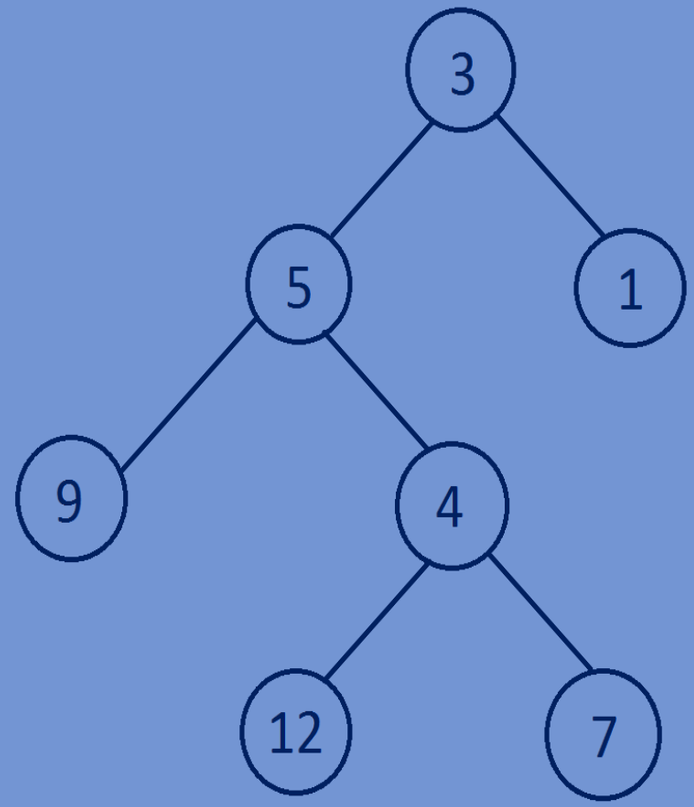
Data Structures

Implementation of a Binary Tree

Representation of a node of a tree:

- A binary tree can have a maximum of two children.
- A node has two links.
- A node is a structure which contains of following parts.
  - Data.
  - Pointer to the left child.
  - Pointer to the right child.

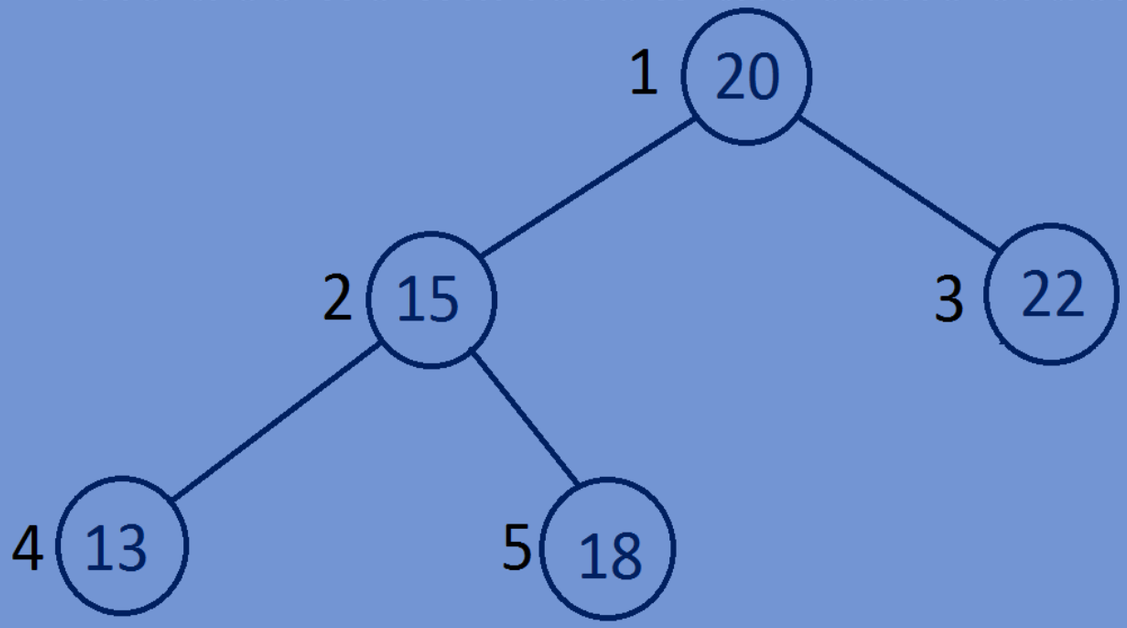


Example of tree node with integer data:

```
typedef struct BTNODE{
    int data;
    struct BTNODE *left;
    struct BTNODE *right;
}BTNODE;
```

Inserting nodes into a Binary Tree:

- Implement complete binary tree.
- Write a insert function that adds a new node in the last level and at the left most Available position.
- Give integer positions to the nodes starting from root node in level order fashion. So that root node is at position 1 and left child of root node will be at position 2 And the right child of the root node will be at position 3,while inserting node into The tree we will follow the same order.



Binary Representation

1-1

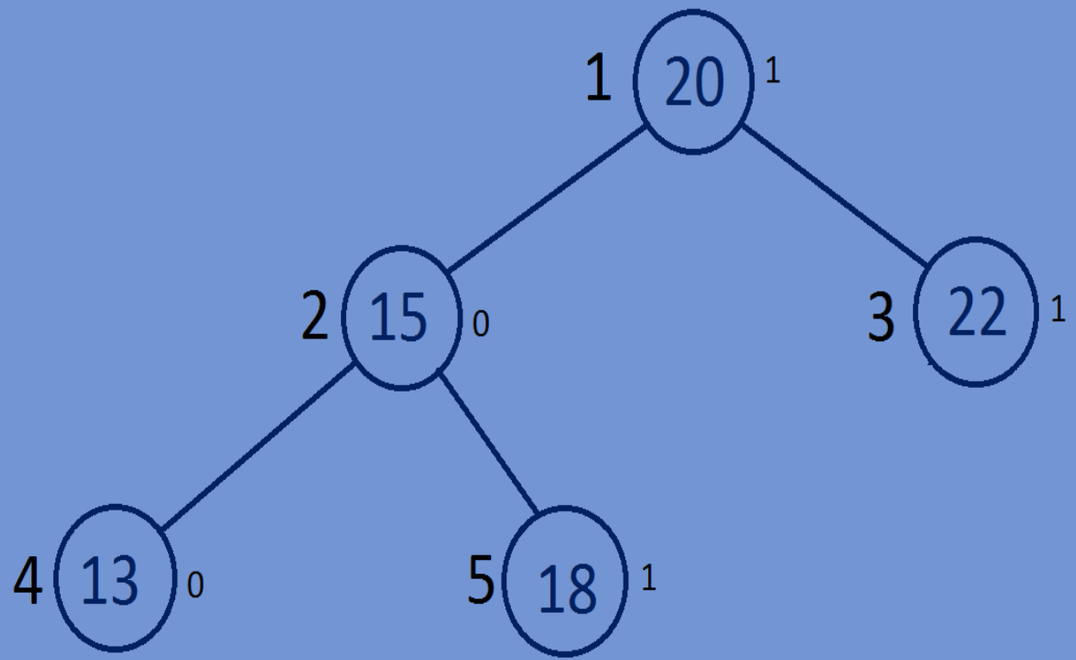
2-10

3-11

4-100

5-101

- For inserting n nodes we traverse through the loop from i=1 to i<=n(including n) And while traversing we take “i” as reference and and if i==1 we insert root node And then we insert new nodes in level order from left to right.
- What we do is we convert the integers into binary no’s and store the binary digits Of each integer In a linked list,now if i==1 we make the new node as root else We traverse the tree using the binary digits=>if we encounter 0 we traverse to the left child else if we encounter 1 we traverse to right child,here we skip the data in first node of the linked list which is always one .



Binary Representation of integers:

1-|1|

2-|1||0|

3-|1||1|

4-|1||0||0|

5-|1||0||1|

Function for converting decimal no into binary digits:

```
//converting a given number into binary digits and storing them into
a linked list
lin_list *binary(int i){
    lin_list *head=NULL;
    while(i!=0){
        lin_list *ptr=(lin_list*)malloc(sizeof(lin_list));
        ptr->data=i%2;
        ptr->next=NULL;
        i=i/2;
        ptr->next=head;
        head=ptr;

    }
    return head;

}
```

Function for inserting nodes into a Binary Tree:

```
//inserting nodes into a tree.
BTNODE *InsertNode(BTNODE *Btree,int n){
    lin_list *Binaryno=NULL;
    for(int i=1;i<=n;i++){
        BTNODE *temp=(BTNODE*)malloc(sizeof(BTNODE)),*temp1=Btree;
        temp->data=i*2;
        temp->left=NULL;
        temp->right=NULL;
        Binaryno=binary(i);
        Binaryno=Binaryno->next;
        if(i==1){
            Btree=temp;
        }
        else{
            while(Binaryno->next){
                if(Binaryno->data==0){
                    temp1=temp1->left;
                }
                else if(Binaryno->data==1){
                    temp1=temp1->right;
                }
```

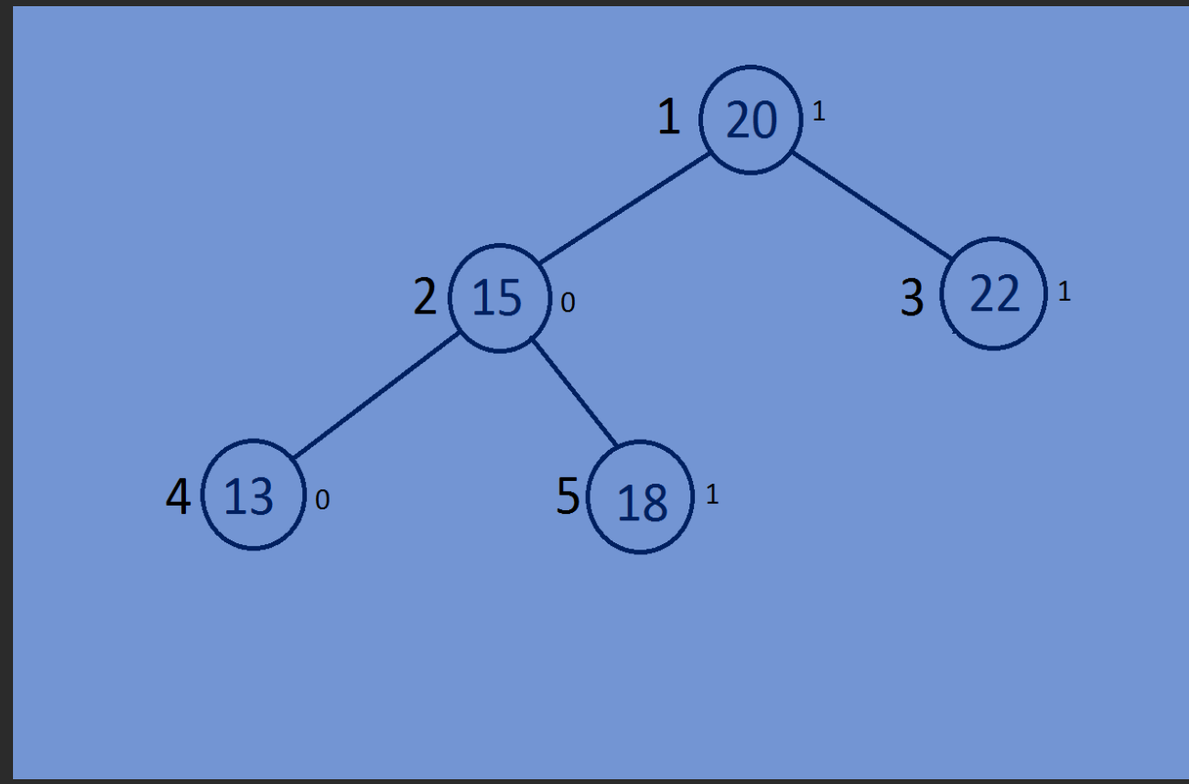
```
                Binaryno=Binaryno->next;
            }
            if(Binaryno->data==0){
                temp1->left=temp;
            }
            else if(Binaryno->data==1){
                temp1->right=temp;
            }
        }
    }
    return Btree;
}
```

Whole program:

```
#include<stdio.h>
#include<stdlib.h>
//creating a node of binary tree.
typedef struct BTNODE{
    int data;
    struct BTNODE *left;
    struct BTNODE *right;
```

```
}BTNODE;
//creating a node for linked list
typedef struct lin_list{
    int data;
    struct lin_list *next;
}lin_list;
//converting a given number into binary digits and storing them into a
linked
lin_list *binary(int i){
    lin_list *head=NULL;
    while(i!=0){
        lin_list *ptr=(lin_list*)malloc(sizeof(lin_list));
        ptr->data=i%2;
        ptr->next=NULL;
        i=i/2;
        ptr->next=head;
        head=ptr;
    }
    return head;
}
//inserting nodes into a tree.
BTNODE *InsertNode(BTNODE *Btree,int n){
    lin_list *Binaryno=NULL;
    for(int i=1;i<=n;i++){
```

```
BTNODE *temp=(BTNODE*)malloc(sizeof(BTNODE)),*temp1=Btree;
temp->data=i*2;
temp->left=NULL;
temp->right=NULL;
Binaryno=binary(i);
Binaryno=Binaryno->next;
if(i==1){
    Btree=temp;
}
else{
    while(Binaryno->next){
        if(Binaryno->data==0){
            temp1=temp1->left;
        }
        else if(Binaryno->data==1){
            temp1=temp1->right;
        }
        Binaryno=Binaryno->next;
    }
    if(Binaryno->data==0){
        temp1->left=temp;
    }
    else if(Binaryno->data==1){
        temp1->right=temp;
```



```
            }
        }
    }
    return Btree;
}
//printing the data of the nodes.
void printdata(BTNODE *Btree){
    if(Btree){
        printf("%d  ",Btree->data);
        printdata(Btree->left);
        printdata(Btree->right);

    }
}
//main function
int main(){
    BTNODE *Btree=NULL;
    int n=10;
    Btree=InsertNode(Btree,n);
    printdata(Btree);
    return 0;
}
```

```
2 4 8 16 18 10 20 6 12 14
```