**Name : Resham Landge**

**Roll No : 2339**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Assignment No : 3** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Title** : Create Binary tree and perform following operations:

a. Insert

b. Display

c. Depth of a tree

d. Display leaf-nodes

e. Create a copy of a tree

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<iostream>

#include<math.h>

#include<string.h>

using namespace std;

int i;

class bintree

{

public:

typedef struct bin

{

char data[10];

struct bin \*left,\*right;

}node;

node \*qu[20],\*p,\*temp,\*newnode,\*root,\*temp1,\*p1,\*root1;

int j,k,front,rare,h;

bintree()

{

root=root1=NULL;

rare=front=-1;

}

int full();

int empty();

void enqueue(node \*p);

node \*deqeue()

{

if(!empty())

{

temp=qu[front++];

p=temp;

return temp;

delete p;

}

cout<<"\n empty \n";

}

void create();

void display(node \*root);

void depth(node \*root);

void disleaf(node \*temp);

void copy1()

{

root1=copy(root);

}

node \*copy(node \*T)

{

node \*temp;

if(T==NULL)

return NULL;

else

{

temp=new node;

strcpy(temp->data,T->data);

temp->left=copy(T->left);

temp->right=copy(T->right);

return temp;

}

}

};

int bintree::full()

{

if(rare==19)

return 1;

return 0;

}

int bintree::empty()

{

if(((rare==-1)&&(front==-1))||(rare<front))

return 1;

return 0;

}

void bintree::enqueue(node \*p)

{

if(!full())

{

if((rare==-1)&&(front==-1))

{

rare=front=0;

qu[rare]=p;

}

else

qu[++rare]=p;

}

else

cout<<"\n queue is full \n";

}

void bintree::create()

{

newnode=new node;

cout<<"\n Enter value ";

cin>>newnode->data;

newnode->left=NULL;

newnode->right=NULL;

if(root==NULL)

{

root=newnode;

enqueue(newnode);

i=1;

}

else

{

if(i==1)

{

temp1=deqeue();

temp1->left=newnode;

enqueue(newnode);

i++;

}

else

{

temp1->right=newnode;

enqueue(newnode);

i=1;

}

}

}

void bintree::display(node \*temp)

{

cout<<"\n display \n";

rare=front=-1;

p=temp;

cout<<"\n\t"<<p->data<<"\n";

enqueue(p);

j=0;

do

{

h=1;

while(h<=pow(2,j))

{

p1=deqeue();

if(p1->left!=NULL)

cout<<"\t"<<p1->left->data;

if(p1->right!=NULL)

cout<<"\t"<<p1->right->data;

enqueue(p1->left);

enqueue(p1->right);

h++;

}

j++;

cout<<"\n";

}while((p1->left!=NULL)&&(p1->right!=NULL));

}

void bintree::depth(node \*root)

{

temp=root;

h=1;

while(temp->left!=NULL)

{

temp=temp->left;

h++;

}

cout<<"\n Depth is "<<h;

}

void bintree::disleaf(node \*temp)

{

if(temp!=NULL)

{

if((temp->left==NULL)&&(temp->right==NULL))

{

cout<<"\n"<<temp->data;

return;

}

disleaf(temp->left);

disleaf(temp->right);

}

}

int main()

{

bintree b1;

char ans,ans1;

int c1,c2;

do

{

cout<<"\n 1-create \n 2-display \n 3-display leaf node \n 4- depath of tree \n 5-copy ";

cout<<"\n Enter ur choice ";

cin>>c1;

switch(c1)

{

case 1:do

{

b1.create();

cout<<"\n do u want to enter more node ";

cin>>ans;

}while(ans=='y');

break;

case 2:b1.display(b1.root);

break;

case 3:b1.disleaf(b1.root);

break;

case 4:b1.depth(b1.root);

break;

case 5:b1.copy1();

cout<<"\n After copy created ";

b1.display(b1.root1);

break;

}

cout<<"\n Do u want to contiune (y/n) ";

cin>>ans1;

}while(ans1=='y');

return 0;

}

Output-

resham@ubuntu:~/Desktop$ g++ ass3.cpp

resham@ubuntu:~/Desktop$ ./a.out

1-create

2-display

3-display leaf node

4- depath of tree

5-copy

Enter ur choice 1

Enter value 1

do u want to enter more node y

Enter value 2

do u want to enter more node y

Enter value 3

do u want to enter more node y

Enter value 4

do u want to enter more node y

Enter value 5

do u want to enter more node n

Do u want to contiune (y/n) y

1-create

2-display

3-display leaf node

4- depath of tree

5-copy

Enter ur choice 2

display

1

2 3

4 5

Do u want to contiune (y/n) y

1-create

2-display

3-display leaf node

4- depath of tree

5-copy

Enter ur choice 3

4

5

Do u want to contiune (y/n) y

1-create

2-display

3-display leaf node

4- depath of tree

5-copy

Enter ur choice 4

Depth is 3

Do u want to contiune (y/n) y

1-create

2-display

3-display leaf node

4- depath of tree

5-copy

Enter ur choice 5

After copy created

display

1

2 3

4 5

Do u want to contiune (y/n) n