package btree;

import java.util.\* ;

public class Btree {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("enter the degree of tree:");

int deg=sc.nextInt();

Tree T=new Tree(deg);

}

}

class Tree{

Node root ;

int degree;

int ht ;

Tree(int d)

{

degree=d ;

root=new Node() ;

root.n=0 ;

root.leaf=true ;

ht=0 ;

Scanner sc=new Scanner(System.in);

System.out.println("enter the number of keys to be inserted: ");

int n=sc.nextInt();

for(int i=1;i<=n;i++)

{

char x=sc.next().charAt(0);

insert(x) ;

display(root) ;

System.out.print("----------------------------------------\n");

}

System.out.print("\n\n ");

display(root) ;

System.out.print("\n HIEGHT : "+ht);

}

public void split(Node x,int i )

{

Node z=new Node() ;

Node y=x.child[i] ;

z.leaf=y.leaf ;

z.n=degree-1 ;

for(int j=1;j<=degree-1;j++)

{

z.keys[j]=y.keys[j+degree];

}

if(!z.leaf)

{

for(int j=1;j<=degree;j++)

{

z.child[j]=y.child[j+degree];

}

}

y.n=degree-1 ;

for(int j=x.n+1;j>=i+1;j--)

{

x.child[j+1]=x.child[j];

}

x.child[i+1]=z ;

for(int j=x.n;j>=i;j--)

{

x.keys[j+1]=x.keys[j];

}

x.keys[i]=y.keys[degree];

x.n=x.n+1 ;

}

public void insert(char k)

{

Node r=root ;

if(r.n==2\*degree-1)

{

Node s=new Node() ;

root =s ;

s.leaf=false ;

s.n=0 ;

s.child[1]=r ;

split(s,1) ;

insert\_non(s,k) ;

ht++ ;

}

else

{

insert\_non(r,k) ;

}

}

public void insert\_non(Node x, char k)

{

int i=x.n ;

if(x.leaf)

{

while(i>=1 && k<x.keys[i])

{

x.keys[i+1]=x.keys[i] ;

i-- ;

}

x.keys[i+1]=k ;

x.n=x.n+1 ;

}

else

{

while(i>=1 && k<x.keys[i])

i-- ;

i++ ;

if( x.child[i].n==2\*degree-1)

{

split(x,i);

if(k>x.keys[i])

i++ ;

}

insert\_non(x.child[i],k) ;

}

}

public void display(Node r)

{

for(int i=1;i<=r.n;i++)

{

System.out.print(r.keys[i]+" ");

}

System.out.print("\n\n ");

for(int i=1;i<=r.n+1;i++)

{

if(r.child[i]!=null)

display(r.child[i]) ;

}

}

}

class Node{

int n ;

char keys[]=new char[10] ;

boolean leaf ;

Node child[]=new Node[10] ;

Node()

{

for(int i=0;i<10;i++)

child[i]=null ;

n=0 ;

}

}

**OUTPUT :**

enter the degree of tree:

3

enter the number of keys to be inserted:

21

F

F

----------------------------------------

S

F S

----------------------------------------

Q

F Q S

----------------------------------------

K

F K Q S

----------------------------------------

C

C F K Q S

----------------------------------------

L

K

/ \

C F L Q S

----------------------------------------

H

K

/ \

C F H L Q S

----------------------------------------

T

K

/ \

C F H L Q S T

----------------------------------------

V

K

/ \

C F H L Q S T V

----------------------------------------

W

K S

/ | \

C F H L Q T V W

----------------------------------------

M

K S

/ | \

C F H L M Q T V W

----------------------------------------

R

K S

/ | \

C F H L MR Q T V W

----------------------------------------

N

K S

/ | \

C F H L MNQR T V W

----------------------------------------

P

K N S

/ | | \

C F H L M P QR T V W

----------------------------------------

A

K N S

/ | | \

A C F H L M P QR T V W

----------------------------------------

B

K N S

/ | | \

A BC F H L M P QR T V W

----------------------------------------

X

K N S

/ | | \

A BC F H L M P QR T V WX

----------------------------------------

Y

K N S

/ | | \

A BC F H L M P QR T V WXY

----------------------------------------

D

C K N S

/ | | | \

A B DF H L M P QR T V WXY

----------------------------------------

Z

C K N S W

/ | | | | \

A B DF H L M P QR T V XY

----------------------------------------

E

C K N S W

/ | | | | \

A B DEF H L M P QR T V XY

----------------------------------------

HIEGHT : 2