Threaded Binary Tree

#include<iostream.h>

#include<conio.h>

#include<process.h>

#define smax 20

#define qmax 30

//keep qmax even

struct node

{ int data;

char thread,visit;

node \*left, \*right;

} \*root=NULL, \*temp, \*t, \*ptr;

node \*q[qmax][2], \*s[smax];

int front,rear,top;

void display(void)

{ ptr=root;

cout<<"Threaded tree (Inorder traversal):\n";

while(1)

{ if(ptr->left==NULL&&ptr->thread=='f')

{ cout<<ptr->data;

goto end;

}

else

{ temp=ptr->left;

if(temp->visit=='n')

{ ptr=ptr->left;

}

else if(ptr->thread=='t')

{ cout<<ptr->data<<" ";

ptr->visit='y';

ptr=ptr->right;

}

else if(ptr->thread=='f')

{ if(ptr->right!=NULL)

{ cout<<ptr->data<<" ";

ptr->visit='y';

ptr=ptr->right;

}

else

{ cout<<ptr->data<<" ";

ptr->visit='y';

goto end;

}

}

}

}

end:

}

void novisit(void)

{ top=0;

s[top]=NULL;

ptr=root;

again:

while(ptr!=NULL)

{ top++;

s[top]=ptr;

ptr=ptr->left;

}

ptr=s[top];

top--;

while(ptr!=NULL)

{ ptr->visit='n';

if((ptr->thread=='f')&&(ptr->right!=NULL))

{ ptr=ptr->right;

goto again;

}

ptr=s[top];

top--;

}

}

void insert(void)

{ char flag='f';

top=0;

temp=new node;

cout<<"\nEnter the data:\t";

cin>>temp->data;

temp->left=NULL;

temp->right=NULL;

temp->thread='f';

temp->visit='n';

if(root==NULL)

{ root=temp;

front=0;

rear=0;

q[rear][0]=temp->left;

q[rear][1]=temp;

if((rear==(qmax-1)&&front==0)||(rear+1)==front)

{ cout<<"\nCan't insert more items.";

goto fin;

}

else

{ if(rear==(qmax-1))

{ rear=0;

}

else

{ rear++;

}

}

q[rear][0]=temp->right;

q[rear][1]=temp;

if((rear==(qmax-1)&&front==0)||(rear+1)==front)

{ cout<<"\nCan't insert more items.";

goto fin;

}

else

{ if(rear==(qmax-1))

{ rear=0;

}

else

{ rear++;

}

}

}

else

{ t=q[front][1];

if(t->left==q[front][0])

{ t->left=temp;

}

else

{ t->right=temp;

}

if((front%2)==1) //right child

{ t->thread='f';

}

if(front==(qmax-1))

{ front=0;

}

else

{ front++;

}

q[rear][0]=temp->left;

q[rear][1]=temp;

if((rear==(qmax-1)&&front==0)||(rear+1)==front)

{ cout<<"\nCan't insert more items.";

goto fin;

}

else

{ if(rear==(qmax-1))

{ rear=0;

}

else

{ rear++;

}

}

q[rear][0]=temp->right;

q[rear][1]=temp;

if((rear==(qmax-1)&&front==0)||(rear+1)==front)

{ cout<<"\nCan't insert more items.";

goto fin;

}

else

{ if(rear==(qmax-1))

{ rear=0;

}

else

{ rear++;

}

}

s[top]=NULL;

ptr=root;

again:

while(ptr!=NULL)

{ top++;

s[top]=ptr;

ptr=ptr->left;

}

if(flag=='t')

{ t=s[top];

goto down;

}

else

{ ptr=s[top];

top--;

while(ptr!=NULL)

{ if(ptr->data==temp->data)

{ flag='t';

}

if((ptr->thread=='f')&&(ptr->right!=NULL))

{ ptr=ptr->right;

goto again;

}

if(flag=='t')

{ t=s[top];

goto down;

}

else

{ ptr=s[top];

top--;

}

}

}

down:

if(t!=NULL)

{ temp->right=t;

temp->thread='t';

}

}

fin:

}

void main()

{ front=-1;

rear=-1;

char ch;

clrscr();

while(1)

{ cout<<"\n1)Insert\n2)Exit\n";

ch=getch();

cout<<ch;

if(ch=='1')

{ insert();

novisit();

display();

}

else if(ch=='2')

{ exit(1);

}

else

{ cout<<"\nWrong choise!!";

}

}

}

