/\*insertion in a b.t as leaf node\*/

#include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node\* left;

struct node\* right;

};

struct node\*new\_root;

struct node\*create\_newnode(struct node\*);

struct node\*insert\_left(struct node\*);

struct node\*display\_left(struct node\*);

struct node\*insert\_right(struct node\*);

struct node\*display\_right(struct node\*);

//creating root node

struct node\*create\_newnode(struct node\*new\_root){

int value;

new\_root= (struct node\*)malloc(sizeof(struct node));

printf("enter your data of root: \n");

scanf("%d",&value);

new\_root->data = value;

new\_root->left = NULL;

new\_root->right = NULL;

return new\_root;

}

int main(){

int op;

create\_newnode(new\_root); //calling the creating func of root node

while(op!=5){

printf("MAIN MENU\n");

printf("choose in where you want to insert a leaf node\n");

printf("1.insert at left\n2.display tree with left child\n3.insert at right\n4.display tree with right child\n5.exit\n");

printf("enter your option: \n");

scanf("%d",&op);

switch(op){

case 1:new\_root=insert\_left(new\_root);

break;

case 2:new\_root=display\_left(new\_root);

break;

case 3:new\_root=insert\_right(new\_root);

break;

case 4:new\_root=display\_right(new\_root);

break;

case 5:exit(0);

default:

printf("invalid choice\n");

}

}

}

//inserting node at left

struct node\*insert\_left(struct node\*new\_root){

int item;

struct node\*ptr,\*new\_leaf;

new\_leaf=(struct node\*)malloc(sizeof(struct node));

printf("enter your data which you want to insert at left: \n");

scanf("%d",&item);

ptr=new\_root;

if(ptr->left==NULL){

new\_leaf->data=item;

new\_root->left=new\_leaf;

new\_leaf->left=NULL;

new\_leaf->right=NULL;

ptr=new\_leaf;

}

printf("node inserted as leaf node at left of root\n");

return new\_root;

}

//displaying the tree after adding a leaf node at left

struct node\*display\_left(struct node\*new\_root){

printf("the tree is below\n");

struct node\*ptr;

ptr=new\_root;

while(ptr!=NULL){

printf("%d->",ptr->data);

ptr=ptr->left;

}

return new\_root;

}

//inserting node at right

struct node\*insert\_right(struct node\*new\_root){

int item;

struct node\*ptr,\*new\_leaf;

new\_leaf=(struct node\*)malloc(sizeof(struct node));

printf("enter your data which you want to insert at left: \n");

scanf("%d",&item);

ptr=new\_root;

if(ptr->right==NULL){

new\_leaf->data=item;

new\_root->right=new\_leaf;

new\_leaf->left=NULL;

new\_leaf->right=NULL;

ptr=new\_leaf;

}

printf("node inserted as leaf node at right of root\n");

return new\_root;

}

//displaying the tree after adding a leaf node at right

struct node\*display\_right(struct node\*new\_root){

printf("the tree is below\n");

struct node\*ptr;

ptr=new\_root;

while(ptr!=NULL){

printf("%d->",ptr->data);

ptr=ptr->right;

}

return new\_root;

}