//AYUSH SAHU

//CSE 11911058

#include<stdio.h>//header files

#include<stdlib.h>//header files

#include<string.h>//header files

typedef struct node

{ char data[100];

struct node\*left;//left pointer

struct node\*right;//right pointer

}Node;

//for creating new bst

Node\*create(char x[100])

{

Node\*temp=(Node\*)malloc(sizeof(Node));

strcpy(temp->data,x);

temp->left=NULL;

temp->right=NULL;

return temp;

}

//for inserting in bst

Node\*insert(Node\*root,char x[100])

{

if(root==NULL)

{

return create(x);

}

if(strcmp(x,root->data)>0)

{

root->right=insert(root->right,x);

}

if(strcmp(x,root->data)<0)

{

root->left=insert(root->left,x);

}

return root;

}

//function for inorder traversal

void inorder(Node\*t)

{

if(t)

{

inorder(t->left);

printf("%s\t",t->data);

inorder(t->right);

}

}

//Driver function

int main()

{

int n,ch;

char m[100];

char str[5][100]={"Arthi","Christy","Dorothy","Fraser","Eliza"};

Node\*root=NULL;

for(int i=0;i<5;i++)

{

root=insert(root,str[i]);

}

printf("bst of 5 name created\n");

for(int i=0;i>=0;i++)

{

printf("\nenter 1 to insert");

printf("\nenter 2 to print:");

printf("\nenter 3 to quit:");

printf("\n\nEnter Your Choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:printf("enter the name to insert\n");

fflush(stdin);

gets(m);

root=insert(root,m);

printf("inserted\n");

break;

case 2:printf("req inorder list is\n");

inorder(root);

break;

case 3:break;

}

}

return 0;

}