```
//Created by Ritwik Chandra Pandey on 2/11/21
//Implementation of splay tree - insertion, inorder traversal



#include<stdio.h>
#include<stdlib.h>



struct node {
    int data;
    struct node *left,*right;
};

typedef struct node * SPLNODE;

SPLNODE root = NULL;

SPLNODE createNodeInSPL(int data) {
        SPLNODE node = (SPLNODE)malloc(sizeof(struct node));
        node->data   = data;
        node->left  = NULL;
        node->right  = NULL;
        return (node);
}

SPLNODE rightRotate(SPLNODE x) {
        SPLNODE y = x->left;
        x->left = y->right;
        y->right = x;
        return y;
}

SPLNODE leftRotate(SPLNODE x) {
        SPLNODE y = x->right;
        x->right = y->left;
        y->left = x;
        return y;
}
```

```
SPLNODE splay(SPLNODE root, int ele) {
	if(root==NULL || root->data == ele)
		return root;

	if(root->data > ele){
		if(root->left == NULL)
			return root;

		if(root->left->data >ele){
			root->left->left = splay (root->left->left, ele);
			root = rightRotate(root);
		}
		else if(root->left->data<ele){
			root->left->right = splay(root->left->right,ele);
			if(root->left->right!=NULL)
				root->left = leftRotate(root->left);
			}
		return (root->left==NULL)? root : rightRotate(root);


	}else{
		if(root->right==NULL) return root;
		if(root->right->data > ele){
			root->right->left = splay(root->right->left,ele);
			if(root->right->left!=NULL)
			  root->right = rightRotate(root->right);
		}
		else if(root->right->data < ele){
			root->right->right = splay(root->right->right,ele);
			root = leftRotate(root);
		}
		return (root->right==NULL)? root: leftRotate(root);
	}
}


SPLNODE insertNodeInSPL(SPLNODE root, int k) {
	if(root == NULL){
		printf("Successfully inserted.\n");
```

```
                return createNodeInSPL(k);
        }
        root = splay(root,k);
        if(root->data == k){
                printf("Element already exists in splay tree.\n");
                return root;
        }
        SPLNODE newnode = createNodeInSPL(k);
        if(root->data > k){
                newnode->right = root;
                newnode->left = root->left;
                root->left = NULL;
        }else{
                newnode->left = root;
                newnode->right = root->right;
                root->right = NULL;
        }
        printf("Successfully inserted.\n");
        return newnode;
}

void inorderInSPL(SPLNODE root) {
        if(root!=NULL){
                inorderInSPL(root->left);
                printf("%d ",root->data);
                inorderInSPL(root->right);
        }
}

void main() {
        int ele, op;
        while(1)
        {
                printf("1.Insert 2.Inorder Traversal 3.Exit\n");
                printf("Enter your option : ");
                scanf("%d", &op);
                switch(op) {
                        case 1:printf("Enter an element to be inserted : ");
                                scanf("%d", &ele);
                                root = insertNodeInSPL(root,ele);
```

```
                                        break;
                                case 2:
                                        if(root == NULL) {
                                                printf("Splay tree is empty.\n");
                                        }
                                        else {
                                                printf("Elements of the AVL tree (in-order traversal): ");
                                                inorderInSPL(root);
                                                printf("\n");
                                        }
                                        break;
                                case 3:exit(0);
                        }
                }
        }
```