```
//Created by Ritwik Chandra Pandey on 2/11/21
//Implementation of splay tree - deletion and preorder 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;
	// Zig-Zig (Left Left)
		if (root->left->data > ele) {
			// First recursively bring the key as root of left-left
			root->left->left = splay(root->left->left, ele);
			// Do first rotation for root, second rotation is done after else
			root = rightRotate(root);
	}
	else if (root->left->data < ele) { // Zig-Zag (Left Right)
			// First recursively bring the key as root of left-right
			root->left->right = splay(root->left->right, ele);
			// Do first rotation for root->left
			if (root->left->right != NULL)
				root->left = leftRotate(root->left);
		}
		// Do second rotation for root
		return (root->left == NULL)? root: rightRotate(root);
	  }
	  else { // Key lies in right subtree
		// Key is not in tree, we are done
		if (root->right == NULL)
			return root;
		// Zig-Zag (Right Left)
		if (root->right->data > ele) {
			// Bring the key as root of right-left
			root->right->left = splay(root->right->left, ele);
	  // Do first rotation for root->right
			if (root->right->left != NULL)
				root->right = rightRotate(root->right);
		}
		else if (root->right->data < ele) {// Zag-Zag (Right Right)
			// Bring the key as root of right-right and do first rotation
			root->right->right = splay(root->right->right, ele);
			root = leftRotate(root);
		}
	// Do second rotation for 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;
}

SPLNODE deleteNodeInSPL (SPLNODE root, int ele) {
        SPLNODE temp;
        if(root==NULL){
                printf("Splay tree is empty.\n");
                return NULL;
        }
        root = splay(root,ele);
        if(root->data!=ele){
                printf("Element not found in the splay tree.\n");
                return root;
```

```
	}
	if(root->left==NULL){
		temp = root;
		root = root->right;
	}else{
		temp = root;
		root = splay(root->left,ele);
		root->right = temp->right;
	}
	printf("Deleted %d from splay tree",temp->data);
	free(temp);
	return root;
}

void preorderInSPL(SPLNODE root) {
	if(root!=NULL){
		printf("%d ",root->data);
		preorderInSPL(root->left);
		preorderInSPL(root->right);
	}
}
void main() {
	int ele, op;
	while(1)
	{
		printf("1.Insert 2.Delete 3.Preorder Traversal 4.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:printf("Enter an element to be deleted : ");
				scanf("%d", &ele);
				root = deleteNodeInSPL(root,ele);
				break;
			case 3:
				if(root == NULL) {
					printf("Splay tree is empty.\n");
```

```
                                        }
                                        else {
                                                printf("Preorder traversal : ");
                                                preorderInSPL(root);
                                                printf("\n");
                                        }
                                        break;
                        case 4:exit(0);
                }
        }
}
```