**Assignment 6**

**Name** : Komal Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

1. **Write a menudriven  program in C to perform following  operations  on TBT**

**a. Create**

**b. Insert**

**c. Inorder traversal**

**d. Preprder traversl**

**e. Postorder traversal**

**h. Exit**

|  |
| --- |
| #include <stdio.h>#include <stdlib.h>struct TBTNode {int lbit, rbit;int data;struct TBTNode\* left, \* right;};struct TBTNode\* create(struct TBTNode\* head, int key);void preorder(struct TBTNode\* head);void Inorder(struct TBTNode\* head);struct TBTNode\* findParent(struct TBTNode\* p);struct TBTNode\* postSuccessor(struct TBTNode\* p);void postorder(struct TBTNode\* head);struct TBTNode\* create(struct TBTNode\* head, int key) {struct TBTNode\* temp, \* p;temp = (struct TBTNode\*)malloc(sizeof(struct TBTNode));temp->data = key;temp->lbit = temp->rbit = 0;if (head->lbit == 0) {head->left = temp;head->lbit = 1;temp->left = temp->right = head;return head;}p = head->left;while (1) {if (key < p->data && p->lbit == 1)p = p->left;else if (key > p->data && p->rbit == 1)p = p->right;elsebreak;}if (key < p->data) {temp->right = p;temp->left = p->left;p->lbit = 1;p->left = temp;}if (key > p->data) {temp->left = p;temp->right = p->right;p->rbit = 1;p->right = temp;}return head;}struct TBTNode\* findParent(struct TBTNode\* p) {struct TBTNode\* child = p;while (p->rbit == 1) p = p->right;p = p->right;if (p->left == child)return p;p = p->left;while (p->right != child) {p = p->right;}return p;}struct TBTNode\* postSuccessor(struct TBTNode\* p) {struct TBTNode\* cur = p;struct TBTNode\* parent = findParent(cur);// printf("suc %d\n", parent->val);if (parent->right == cur) return parent;else {while (p->rbit == 1) p = p->right;p = p->right;if (p->rbit == 1) {p = p->right;while (!(p->rbit == 0 && p->lbit == 0)) {if (p->lbit == 1) p = p->left;else if (p->rbit == 1) p = p->right;}}}return p;}void postorder(struct TBTNode\* head) {struct TBTNode\* p = head->left;struct TBTNode\* temp = p;while (!(p->rbit == 0 && p->lbit == 0)) {if (p->lbit == 1) {p = p->left;}else if (p->rbit == 1) p = p->right;}printf(" %d", p->data);while (p != head->left) {p = postSuccessor(p);printf(" %d", p->data);}}void preorder(struct TBTNode\* head) {struct TBTNode\* p;p = head->left;while (p != head) {printf("%d\n", p->data);if (p->lbit == 1)p = p->left;else if (p->rbit == 1)p = p->right;else if (p->rbit == 0) {while (p->rbit == 0)p = p->right;p = p->right;}}}void Inorder(struct TBTNode\* head) {struct TBTNode\* p;p = head->left;while (p != head) {while (p->lbit == 1)p = p->left;printf("%d\n", p->data);while (p->rbit == 0 || p->right == head) {p = p->right;if (p == head)break;printf("%d\n", p->data);}p = p->right;}}int main() {struct TBTNode\* head = NULL;int n, key;head = (struct TBTNode\*)malloc(sizeof(struct TBTNode));head->rbit = 1;head->lbit = 0;head->left = head->right = head;printf("No. of nodes:");scanf("%d", &n);for (int i = 0; i < n; i++) {printf("Enter Value:");scanf("%d", &key);head = create(head, key);}printf("Inorder Traversal:\n");Inorder(head);printf("Preorder Traversal:\n");preorder(head);printf("Postorder Traversal:\n");postorder(head);return 0;} |

**Output:**

