**GROUP D**

**A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword**

|  |  |
| --- | --- |
|  | #include<iostream> |
|  | #include<stdio.h> |
|  | #include<string.h> |
|  | using namespace std; |
|  | class Tree |
|  | { |
|  | typedef struct node |
|  | { |
|  | char key[10]; |
|  | char meaning[10]; |
|  | struct node \*left; |
|  | struct node \* right; |
|  | }btree; |
|  | public: |
|  | btree \*New,\*root; |
|  | Tree(); |
|  | void create(); |
|  | void insert(btree \*root,btree \*New); |
|  | void inorder(); |
|  | void inorder\_rec(btree \*root); |
|  | void postorder(); |
|  | void postorder\_rec(btree \*root); |
|  | }; |
|  | Tree::Tree() |
|  | { |
|  | root=NULL; |
|  | } |
|  | void Tree::inorder() |
|  | { |
|  | inorder\_rec(root); |
|  | } |
|  | void Tree::inorder\_rec(btree \*root) |
|  | { |
|  | if(root!=NULL) |
|  | { |
|  | inorder\_rec(root->left); |
|  | cout<<"\n\t"<<root->key<<"\t"<<root->meaning; |
|  | inorder\_rec(root->right); |
|  | } |
|  | } |
|  | void Tree::postorder() |
|  | { |
|  | postorder\_rec(root); |
|  | } |
|  | void Tree::postorder\_rec(btree \*root) |
|  | { |
|  | if(root!=NULL) |
|  | { |
|  |  |
|  | postorder\_rec(root->right); |
|  | cout<<"\n\t"<<root->key<<"\t"<<root->meaning; |
|  | postorder\_rec(root->left); |
|  | } |
|  | } |
|  | void Tree::create() |
|  | { |
|  | New=new btree; |
|  | New->left=New->right=NULL; |
|  | cout<<"\n\tEnter the Keyword: "; |
|  | cin>>New->key; |
|  | cout<<"\n\tEnter the Meaning of "<<New->key<<" : "; |
|  | cin>>New->meaning; |
|  | if(root==NULL) |
|  | { |
|  | root=New; |
|  | } |
|  | else |
|  | { |
|  | insert(root,New); |
|  | } |
|  | } |
|  | void Tree::insert(btree \*root,btree \*New) |
|  | { |
|  |  |
|  | if(strcmp(root->key,New->key)>0) |
|  | { |
|  | if(root->left==NULL) |
|  | root->left=New; |
|  | else |
|  | insert(root->left,New); |
|  | } |
|  | else |
|  | { |
|  | if(root->right==NULL) |
|  | root->right=New; |
|  | else |
|  | insert(root->right,New); |
|  | } |
|  | } |
|  |  |
|  | main() |
|  | { |
|  | Tree tr; |
|  | int ch; |
|  | char ans; |
|  | do |
|  | { |
|  | cout<<"\n\t\*\*\*\*\* BST Operations \*\*\*\*\*"; |
|  | cout<<"\n\t1. Create\n\t2. Display\n\t3. Exit"; |
|  | cout<<"\n\t.....Enter Your Choice: "; |
|  | cin>>ch; |
|  | switch(ch) |
|  | { |
|  | case 1: |
|  |  |
|  | do |
|  | { |
|  | tr.create(); |
|  | cout<<"......Do You Want To Continue: "; |
|  | cin>>ans; |
|  | }while(ans=='y'||ans=='Y'); |
|  | break; |
|  | case 2: cout<<"\n\t\t1. Ascending\n\t\t2. Descending\n\t\t.....Enter Your Choice: "; |
|  | cin>>ch; |
|  | cout<<"\n\tKeyword\tMeaning"; |
|  |  |
|  | switch(ch) |
|  | { |
|  | case 1: |
|  | tr.inorder(); |
|  | break; |
|  | case 2: |
|  | tr.postorder(); |
|  | break; |
|  | } |
|  | break; |
|  | case 3: |
|  | break; |
|  | } |
|  | cout<<"\n\t\t..... Do You Want to Continue: "; |
|  | cin>>ans; |
|  | }while(ans=='y'||ans=='Y'); |
|  | } |
|  |  |
|  | /\* |
|  | siem@siem-OptiPlex-3010:~$ cd Desktop/ |
|  | siem@siem-OptiPlex-3010:~/Desktop$ g++ ass2.cpp |
|  | siem@siem-OptiPlex-3010:~/Desktop$ ./a.out |
|  |  |
|  | \*\*\*\*\* BST Operations \*\*\*\*\* |
|  | 1. Create |
|  | 2. Display |
|  | 3. Exit |
|  | .....Enter Your Choice: 1 |
|  |  |
|  | Enter the Keyword: int |
|  |  |
|  | Enter the Meaning of int : datatype |
|  | ......Do You Want To Continue: y |
|  |  |
|  | Enter the Keyword: for |
|  |  |
|  | Enter the Meaning of for : loop |
|  | ......Do You Want To Continue: y |
|  |  |
|  | Enter the Keyword: if |
|  |  |
|  | Enter the Meaning of if : condition |
|  | ......Do You Want To Continue: y |
|  |  |
|  | Enter the Keyword: < |
|  |  |
|  | Enter the Meaning of < : lessthan |
|  | ......Do You Want To Continue: y |
|  |  |
|  | Enter the Keyword: malloc |
|  |  |
|  | Enter the Meaning of malloc : memory |
|  | ......Do You Want To Continue: y |
|  |  |
|  | Enter the Keyword: while |
|  |  |
|  | Enter the Meaning of while : loop |
|  | ......Do You Want To Continue: n |
|  |  |
|  | ..... Do You Want to Continue: y |
|  |  |
|  | \*\*\*\*\* BST Operations \*\*\*\*\* |
|  | 1. Create |
|  | 2. Display |
|  | 3. Exit |
|  | .....Enter Your Choice: 2 |
|  |  |
|  | 1. Ascending |
|  | 2. Descending |
|  | .....Enter Your Choice: 1 |
|  |  |
|  | Keyword Meaning |
|  | < lessthan |
|  | for loop |
|  | if condition |
|  | int datatype |
|  | malloc memory |
|  | while loop |
|  | ..... Do You Want to Continue: y |
|  |  |
|  | \*\*\*\*\* BST Operations \*\*\*\*\* |
|  | 1. Create |
|  | 2. Display |
|  | 3. Exit |
|  | .....Enter Your Choice: 2 |
|  |  |
|  | 1. Ascending |
|  | 2. Descending |
|  | .....Enter Your Choice: 2 |
|  |  |
|  | Keyword Meaning |
|  | while loop |
|  | malloc memory |
|  | int datatype |
|  | if condition |
|  | for loop |
|  | < lessthan |
|  | ..... Do You Want to Continue: |
|  | \*/ |