

**21-april-2021**

**2019115126**

**Swaminathan Navinashok**

1. Create an AVL Tree to store the characters/alphabets by strictly following AVL properties. Write suitable routines for insertion, deletion and display operations. In addition, write suitable routines for the following operations too:
  - a. Find the grandparent of the node having maximum value in AVL tree.
  - b. Find the parent's sibling for the node having minimum value in AVL tree.
  - c. Display the longest path in the tree possessing maximum height.
  - d. Difference between the height of left sub tree and right sub tree of the given node. (To check whether it is balanced)
  - e. Returning the address of the parent and grandparent once the node gets deleted.

## **Main.cpp**

```
#include "AVL.h"  
  
#include<iostream>  
  
#include<vector>  
  
using namespace std;  
  
  
int main() {  
  
    AVL<char> at;  
  
    int option;int n;  
  
    char val;int hleft=-1,hright=-1;  
  
    vector<char> output;
```

```

vector<char> chi;char aunt;

do {

    cout << "\n\n\n What operation do you want to perform? Select Option number" << endl;
    cout << "\n 1. Insert Node" << endl;
    cout << "\n 2. Delete Node and return its parent and grandparent" << endl;
    cout << "\n 3. display" << endl;
    cout << "\n 4. Height of Tree and display that longest path path" << endl;
    cout << "\n 5. find min of Tree and it's parent's sibling"" << endl;
    cout << "\n 6. find max of Tree and its grandpa" << endl;
    cout << "\n 7. find Node" << endl;
    cout << "\n 0. Exit Program\n\n  ";

    cout<<"  ";cin >> option;

switch (option) {

case 0:
    break;

case 1:
    cout << "\n\n AVL INSERT" << endl;
    cout << "\n\n Enter VALUE of TREE NODE to INSERT in AVL Tree:  ";
    cin >> val;
    at.insert(val);
    break;

case 2:
    cout << "\n\n DELETE" << endl;
}

```

```

cout << "\n\n    Enter VALUE of TREE NODE to DELETE in AVL:   ";
cin >> val;

output = at.parent(val,at.root);

n= output.size();

if(n==0)

{

    cout<< "\n\n      root has no parent or grandparent ";

}

else if(n==1)

{

    cout<<"\n\n      no grand parent";cout<<" \n\n      parent : "<<output[n-1];

}

else

    cout<<" \n\n      parent : "<<output[n-1]<<"\n\n      grandparent : "<<output[n-2];



at.remove(val);

break;

case 3:

at.display();

break;

case 4:

cout << "\n\n height of tree   ";

```

```
cout<<at.height(at.root);

output = at.longestPath(at.root);

n= output.size();

cout<<" \n\n longest path:\n\n      ";

cout << output[n - 1];

for (int i = n - 2; i >= 0; i--) {

    cout << " -> " << output[i];

}

break;
```

case 5:

```
cout << "\n\n      minimum   ";val=at.getdata(at.findMin(at.root));

cout<<val;

output = at.parent(val,at.root);

n= output.size();

chi = at.child(val,at.root);

n= chi.size();

if(chi[n-1]!=output[n-1])

    aunt=chi[n-1];

else aunt=chi[n-2];



if(aunt=='#')

{
```

```

cout<<"\n\n    parent has no sibling ";
}

else

{
    cout<<"\n\n    parent's sibling' : "<<aunt;
}

break;

```

case 6:

```

cout << "\n\n    maximum    ";
val=at.getdata(at.findMax(at.root));

cout<<val;
output = at.parent(val,at.root);
n= output.size();
if(n==0)

{
    cout<< "\n\n    root has no parent or grandparent ";
}

else if(n==1)

{
    cout<<"\n\n    no grandparent";
}

else

```

```
cout<<"\n\n    grandparent : "<<output[n-2];  
  
break;
```

case 7:

```
cout << "\n\n Enter VALUE to find in AVL Tree: ";
```

```
cin >> val;
```

```
if(at.find(val,at.root))
```

```
{
```

```
    cout<< "\n\n    present ";
```

```
}
```

```
else{
```

```
    cout<< "\n\n    absent ";
```

```
}
```

```
break;
```

default:

```
cout << "\n\n    Enter Proper Option number " << endl;
```

```
}
```

```
} while (option != 0);
```

```
return 0;
```

```
}
```

## AVL.h

```
#ifndef AVL_H
```

```
#define AVL_H
```

```
#include<iostream>
```

```
#include<vector>
```

```
using namespace std;
```

```
template <typename T>
```

```
class AVL
```

```
{
```

```
public:
```

```
struct node
```

```
{
```

```
    T data='#';
```

```
    node* left;
```

```
    node* right;
```

```
    int height=-1;
```

```
};
```

```
node* root;
```

```
void preorder(node* p);

vector<T> longestPath(node* root);
vector<T> parent(T val,node* root);

vector<T> child(T val,node* root);

node* addressofval(T val,node* t);

void postorder(node* p);

void makeEmpty(node* t);

node* insert(const T& x, node* t);

T find(const T& x, node* t);

node* singleRightRotate(node* &t);

node* singleLeftRotate(node* &t);

node* doubleLeftRotate(node* &t);

node* doubleRightRotate(node* &t);

node* findMin(node* t);
```

```
node* findMax(node* t);

node* remove(T x, node* t);

int height(node*t);

T getBalance(node* t);

void inorder(node* t);

AVL();

void insert(T x);

void remove(T x);

void display();

T getdata(node *t);

};

#endif
```

## **AVL.cpp**

```
#include "AVL.h"

#include<vector>
#include<iostream>
using namespace std;

template class AVL<char>;
template class AVL<int>;

template <typename T>
void AVL<T>::preorder(typename AVL<T>::node* p)
{
    if (p!=NULL)
    {
        cout<<p->data<<" ";
        preorder(p->left);
        preorder(p->right);
    }
}

template <typename T>
```

```
void AVL<T>:: postorder(typename AVL<T>::node* p)
{
    if (p!=NULL)
    {
        postorder(p->left);
        postorder(p->right);
        cout<<p->data<<" ";
    }
}
```

```
template <typename T>
void AVL<T>:: makeEmpty(typename AVL<T>::node* t)
{
    if(t == NULL)
        return;
    makeEmpty(t->left);
    makeEmpty(t->right);
    t=NULL;
    delete t;
}
```

```
template <typename T>
typename AVL<T>::node* AVL<T>:: insert(const T& x, typename AVL<T>::node* t)
{
    if(t == NULL)
    {
        t = new typename AVL<T>::node;
        t->data = x;
        t->height = 0;
    }
}
```

```

t->left = t->right = NULL;
}

else if(x < t->data)

{
    t->left = insert(x, t->left);

    if(height(t->left) - height(t->right) == 2)

    {
        if(x < t->left->data)

            t = singleRightRotate(t);

        else

            t = doubleRightRotate(t);

    }

}

else if(x > t->data)

{
    t->right = insert(x, t->right);

    if(height(t->right) - height(t->left) == 2)

    {
        if(x > t->right->data)

            t = singleLeftRotate(t);

        else

            t = doubleLeftRotate(t);

    }

}

t->height = max(height(t->left), height(t->right))+1;

return t;
}

```

```
template <typename T>
T AVL<T>::find(const T& x, typename AVL<T>::node* t)
{
    if(t == NULL)
    {
        return 0;
    }

    else if(x < t->data)
    {
        return find(x, t->left);
    }

    else if(x > t->data)
    {
        return find(x, t->right);
    }

    else
    {
        return 1;
    }
}
```

```
template <typename T>
```

```

typename AVL<T>::node* AVL<T>::singleRightRotate(typename AVL<T>::node* &t)

{
    if (t->left != NULL) {

        typename AVL<T>::node* u = t->left;

        t->left = u->right;

        u->right = t;

        t->height = max(height(t->left), height(t->right)) + 1;

        u->height = max(height(u->left), t->height) + 1;

        return u;

    }

    return t;
}

```

```

template <typename T>

typename AVL<T>::node* AVL<T>::singleLeftRotate(typename AVL<T>::node* &t)

{
    if (t->right != NULL) {

        typename AVL<T>::node* u = t->right;

        t->right = u->left;

        u->left = t;

        t->height = max(height(t->left), height(t->right)) + 1;

        u->height = max(height(t->right), t->height) + 1;

        return u;

    }

    return t;
}

```

```

template <typename T>
    typename AVL<T>::node* AVL<T>::doubleLeftRotate(typename AVL<T>::node* &t)
{
    t->right = singleRightRotate(t->right);
    return singleLeftRotate(t);
}

template <typename T>
    typename AVL<T>::node* AVL<T>::doubleRightRotate(typename AVL<T>::node* &t)
{
    t->left = singleLeftRotate(t->left);
    return singleRightRotate(t);
}

template <typename T>
    typename AVL<T>::node* AVL<T>::findMin(typename AVL<T>::node* t)
{
    if(t == NULL)
        return NULL;
    else if(t->left == NULL)
        return t;
    else
        return findMin(t->left);
}

template <typename T>
    typename AVL<T>::node* AVL<T>::findMax(typename AVL<T>::node* t)
{
    if(t == NULL)

```

```

        return NULL;

    else if(t->right == NULL)

        return t;

    else

        return findMax(t->right);

    }

template <typename T>

typename AVL<T>::node* AVL<T>::remove(T x, typename AVL<T>::node* t)

{

    typename AVL<T>::node* temp;

    if(t == NULL)

        return NULL;

    else if(x < t->data)

        t->left = remove(x, t->left);

    else if(x > t->data)

        t->right = remove(x, t->right);

    else if(t->left && t->right)

    {

        temp = findMin(t->right);

        t->data = temp->data;

        t->right = remove(t->data, t->right);

    }

}

```

```

    }

else
{
    temp = t;
    if(t->left == NULL)
        t = t->right;
    else if(t->right == NULL)
        t = t->left;
    delete temp;
}

if(t == NULL)
    return t;

t->height = max(height(t->left), height(t->right))+1;

if(height(t->left) - height(t->right) == 2)
{
    if(height(t->left->left) - height(t->left->right) == 1)
        return singleLeftRotate(t);

    else
        return doubleLeftRotate(t);
}

else if(height(t->right) - height(t->left) == 2)
{

```

```
    if(height(t->right->right) - height(t->right->left) == 1)
        return singleRightRotate(t);

    else
        return doubleRightRotate(t);
    }

    return t;
}
```

```
template <typename T>

int AVL<T>::height(typename AVL<T>::node* t)

{
    return (t == NULL ? -1 : t->height);
}
```

```
template <typename T>

T AVL<T>::getBalance(typename AVL<T>::node* t)

{
    if(t == NULL)
        return 0;
    else
        return height(t->left) - height(t->right);
}
```

```
template<typename T>

void AVL<T>::inorder(typename AVL<T>::node* t)

{
```

```
if(t == NULL)
    return;
inorder(t->left);
cout << t->data << " ";
inorder(t->right);
}
```

```
template<typename T>
AVL<T>::AVL()
{
    root = NULL;
}
```

```
template <typename T>
void AVL<T>:: insert(T x)
{
    root = insert(x, root);
}
```

```
template <typename T>
void AVL<T>:: remove(T x)
{
    root = remove(x, root);
}
```

```
template <typename T>
void AVL<T>::display()
{
    cout<<"\n\n inorder traversal :\n      ";
    inorder(root);
    cout<<"\n\n preorder traversal :\n      ";
    preorder(root);
    cout<<"\n\n postorder traversal :\n      ";
    postorder(root);
    cout << endl;
}
```

```
template <typename T>
T AVL<T>::getdata(AVL<T>::node *t)
{
    return t->data;
}
```

```
template<typename T>
vector<T> AVL<T>::longestPath(typename AVL<T>::node* root)
{
    if (root == NULL) {
        vector<T> temp
        = {};
    }
```

```
    return temp;
}

vector<T> rightvect
= longestPath(root->right);

vector<T> leftvect
= longestPath(root->left);

if (leftvect.size() > rightvect.size())
    leftvect.push_back(root->data);
else
    rightvect.push_back(root->data);

return (leftvect.size() > rightvect.size()
? leftvect
: rightvect);
}

template<typename T>
vector<T> AVL<T>::parent(T val,typename AVL<T>::node* root)
```

```
{  
  
if (root == NULL) {  
    vector<T> temp  
    = {};  
    return temp;  
}  
  
vector<T> parentvect={};typename AVL<T>::node* t;t=root;  
  
while(t->data!=val)  
{  
    parentvect.push_back(t->data);  
    if(t->data>val)  
        t=t->left;  
    else if (t->data<val)  
        t=t->right;  
}  
return parentvect;
```

```
}
```

```
template<typename T>  
vector<T> AVL<T>::child(T val,typename AVL<T>::node* root)  
{
```

```
if (root == NULL) {
```

```
    vector<T> temp
```

```
    = {};
```

```
    return temp;
```

```
}
```

```
vector<T> parentvect={};typename AVL<T>::node* t;t=root;
```

```
while(t->data!=val)
```

```
{
```

```
    if(t->data>val)
```

```
        t=t->left;
```

```
    else if (t->data<val)
```

```
        t=t->right;
```

```

    }

    if(t->left!=NULL)
    {
        parentvect.push_back(t->left->data);
    }
    else
    {
        parentvect.push_back('#');

    }

    if(t->right!=NULL)
    {
        parentvect.push_back(t->right->data);
    }
    else
    {
        parentvect.push_back('#');

    }

    return parentvect;
}

template<typename T>
typename AVL<T>::node* AVL<T>::adressofval(T val,typename AVL<T>::node* t)

```

```
{\n    if(val>t->data)\n    {\n        return addressofval(val,t->right);\n    }\n    else if(val<t->data)\n    {\n        return addressofval(val,t->left);\n    }\n    else if(val==t->data)\n    {\n        return t;\n    }\n    else\n        return NULL;\n}
```

Project1 - [Project1dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

427 }
428 }
429
430 template<typename T>
431 typename AVL<T>::node* AVL<T>::adressofval(T val,typename AVL<T>::node* t)
432 {
433     if(val>t->data)
434     {
435         return adressofval(val,t->right);
436     }
437     else if(val<t->data)
438     {
439         return adressofval(val,t->left);
440     }
441     else if(val==t->data)
442     {
443         return t;
444     }
445     else
446     {
447         return NULL;
448     }
449
450 }
```

Compiler Resources Compile Log Debug Find Results (27) Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Project1 - [Project1dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

412 }
413 }
414 }
415 if(t->right!=NULL)
416 {
417     parentvect.push_back(t->right->data);
418 }
419 else
420 {
421     parentvect.push_back('#');
422 }
423
424 return parentvect;
425
426
427
428
429
430 template<typename T>
431 typename AVL<T>::node* AVL<T>::adressofval(T val,typename AVL<T>::node* t)
432 {
433     if(val>t->data)
434     {
435         return adressofval(val,t->right);
436     }
437 }
```

Compiler Resources Compile Log Debug Find Results (27) Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

392     while(t->data!=val)
393     {
394         if(t->data>val)
395             t=t->left;
396         else if (t->data<val)
397             t=t->right;
398     }
399     if(t->left!=NULL)
400     {
401         parentvect.push_back(t->left->data);
402     }
403     else
404     {
405         parentvect.push_back('#');
406     }
407     if(t->right!=NULL)
408     {
409     }

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

377     template<typename T>
378     vector<T> AVL<T>::child(T val,typename AVL<T>::node* root)
379     {
380         if (root == NULL) {
381             vector<T> temp
382             = {};
383             return temp;
384         }
385         vector<T> parentvect={};typename AVL<T>::node* t;t=root;
386
387         while(t->data!=val)
388         {
389             if(t->data>val)
390                 t=t->left;
391             else if (t->data<val)
392                 t=t->right;
393         }
394     }

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

357
358
359
360     while(t->data!=val)
361     {
362         parentvect.push_back(t->data);
363         if(t->data>val)
364             | t=t->left;
365         else if (t->data<val)
366             | t=t->right;
367
368
369     }
370
371
372
373
374 }
375
376
377 template<typename T>
378 vector<T> AVL<T>::child(T val,typename AVL<T>::node* root)
379 {
380 }
```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

341
342
343     template<typename T>
344     vector<T> AVL<T>::parent(T val,typename AVL<T>::node* root)
345     {
346
347
348         if (root == NULL) {
349             vector<T> temp
350             | = {};
351             return temp;
352         }
353
354
355         vector<T> parentvect={};typename AVL<T>::node* t;t=root;
356
357
358
359
360         while(t->data!=val)
361         {
362             parentvect.push_back(t->data);
363             if(t->data>val)
364                 | t=t->left;
365         }
366
367
368
369     }
```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

329     leftvect.push_back(root->data);
330
331     else
332         rightvect.push_back(root->data);
333
334
335     return (leftvect.size() > rightvect.size()
336             ? leftvect
337             : rightvect);
338 }

339
340
341
342
343     template<typename T>
344     vector<T> AVL<T>::parent(T val,typename AVL<T>::node* root)
345     {
346
347         if (root == NULL) {
348             vector<T> temp
349             = {};
350             return temp;
351         }
352     }

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

306
307
308     template<typename T>
309     vector<T> AVL<T>::longestPath(typename AVL<T>::node* root)
310     {
311
312         if (root == NULL) {
313             vector<T> temp
314             = {};
315             return temp;
316         }
317
318
319         vector<T> rightvect
320             = longestPath(root->right);
321
322
323         vector<T> leftvect
324             = longestPath(root->left);
325
326
327         if (leftvect.size() > rightvect.size())
328             leftvect.push_back(root->data);
329     }

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

286     template <typename T>
287     void AVL<T>::display()
288     {
289         cout<<"\n\n inorder traversal : \n      ";
290         inorder(root);
291         cout<<"\n\n preorder traversal : \n      ";
292         preorder(root);
293         cout<<"\n\n postorder traversal : \n      ";
294         postorder(root);
295         cout << endl;
296     }
297
298
299
300     template <typename T>
301     T AVL<T>::getdata(AVL<T>::node *t)
302     {
303         return t->data;
304     }
305
306
307
308     template<typename T>
309     vector<T> AVL<T>::longestPath(typename AVL<T>::node* root)

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

273     template <typename T>
274     void AVL<T>:: insert(T x)
275     {
276         root = insert(x, root);
277     }
278
279
280     template <typename T>
281     void AVL<T>:: remove(T x)
282     {
283         root = remove(x, root);
284     }
285
286
287     template <typename T>
288     void AVL<T>::display()
289     {
290         cout<<"\n\n inorder traversal : \n      ";
291         inorder(root);
292         cout<<"\n\n preorder traversal : \n      ";
293         preorder(root);
294         cout<<"\n\n postorder traversal : \n      ";
295         postorder(root);
296         cout << endl;
297     }

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

255     template<typename T>
256     void AVL<T>::inorder(typename AVL<T>::node* t)
257     {
258         if(t == NULL)
259             return;
260         inorder(t->left);
261         cout << t->data << " ";
262         inorder(t->right);
263     }
264
265     template<typename T>
266     AVL<T>::AVL()
267     {
268         root = NULL;
269     }
270
271
272     template <typename T>
273     void AVL<T>::insert(T x)
274     {
275         root = insert(x, root);
276     }
277
278

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

238     template <typename T>
239     int AVL<T>::height(typename AVL<T>::node* t)
240     {
241         if(t == NULL)
242             return (-1);
243         else
244             return (t->left->height + 1);
245     }
246
247     template <typename T>
248     T AVL<T>::getBalance(typename AVL<T>::node* t)
249     {
250         if(t == NULL)
251             return 0;
252         else
253             return height(t->left) - height(t->right);
254     }
255
256     template<typename T>
257     void AVL<T>::inorder(typename AVL<T>::node* t)
258     {
259         if(t == NULL)
260             return;
261         inorder(t->left);
262         cout << t->data << " ";
263     }

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

221     return singleLeftRotate(t);
222
223     else
224         return doubleLeftRotate(t);
225
226
227     else if(height(t->right) - height(t->left) == 2)
228     {
229
230         if(height(t->right->right) - height(t->right->left) == 1)
231             return singleRightRotate(t);
232
233         else
234             return doubleRightRotate(t);
235
236     }
237
238
239     template <typename T>
240     int AVL<T>::height(typename AVL<T>::node* t)
241     {
242         return (t == NULL ? -1 : t->height);
243     }
244

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

201
202
203     else
204     {
205         temp = t;
206         if(t->left == NULL)
207             t = t->right;
208         else if(t->right == NULL)
209             t = t->left;
210         delete temp;
211     }
212     if(t == NULL)
213         return t;
214
215     t->height = max(height(t->left), height(t->right))+1;
216
217
218     if(height(t->left) - height(t->right) == 2)
219     {
220
221         if(height(t->left->left) - height(t->left->right) == 1)
222             return singleLeftRotate(t);
223
224         else
225             return doubleLeftRotate(t);
226
227
228     }
229
230
231     template <typename T>
232     int AVL<T>::height(typename AVL<T>::node* t)
233     {
234         return (t == NULL ? -1 : t->height);
235     }
236

```

Compiler Resources Compile Log Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

template <typename T>
typename AVL<T>::node* AVL<T>::remove(T x, typename AVL<T>::node* t)
{
    typename AVL<T>::node* temp;

    if(t == NULL)
        return NULL;

    else if(x < t->data)
        t->left = remove(x, t->left);
    else if(x > t->data)
        t->right = remove(x, t->right);

    else if(t->left == t->right)
    {
        temp = findMin(t->right);
        t->data = temp->data;
        t->right = remove(t->data, t->right);
    }
}

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

template <typename T>
typename AVL<T>::node* AVL<T>::findMin(typename AVL<T>::node* t)
{
    if(t == NULL)
        return NULL;
    else if(t->left == NULL)
        return t;
    else
        return findMin(t->left);
}

template <typename T>
typename AVL<T>::node* AVL<T>::findMax(typename AVL<T>::node* t)
{
    if(t == NULL)
        return NULL;
    else if(t->right == NULL)
        return t;
    else
        return findMax(t->right);
}

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

141     template <typename T>
142     typename AVL<T>::node* AVL<T>::doubleLeftRotate(typename AVL<T>::node* &t)
143     {
144         t->right = singleRightRotate(t->right);
145         return singleLeftRotate(t);
146     }
147
148
149     template <typename T>
150     typename AVL<T>::node* AVL<T>::doubleRightRotate(typename AVL<T>::node* &t)
151     {
152         t->left = singleLeftRotate(t->left);
153         return singleRightRotate(t);
154     }
155
156     template <typename T>
157     typename AVL<T>::node* AVL<T>::findMin(typename AVL<T>::node* t)
158     {
159         if(t == NULL)
160             return NULL;
161         else if(t->left == NULL)
162             return t;
163         else
164             return findMin(t->left);
165     }

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

125
126
127     template <typename T>
128     typename AVL<T>::node* AVL<T>::singleLeftRotate(typename AVL<T>::node* &t)
129     {
130         if (t->right != NULL) {
131             typename AVL<T>::node* u = t->right;
132             t->right = u->left;
133             u->left = t;
134             t->height = max(height(t->left), height(t->right)) + 1;
135             u->height = max(height(t->right), t->height) + 1;
136             return u;
137         }
138         return t;
139     }
140
141
142     template <typename T>
143     typename AVL<T>::node* AVL<T>::doubleLeftRotate(typename AVL<T>::node* &t)
144     {
145         t->right = singleRightRotate(t->right);
146         return singleLeftRotate(t);
147     }
148

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

16:39 22-04-2021 ENG

The image shows two instances of the Dev-C++ IDE running side-by-side. Both windows have the title "Project1 - [Project1.dev] - Dev-C++ 5.11".

**Top Window (Left):**

- Project Explorer:** Shows a project named "Project1" with files: main.cpp, AVL.h, and AVL.cpp.
- Code Editor:** Displays the implementation of AVL tree rotations. The code includes two templates: `singleRightRotate` and `singleLeftRotate`. The `singleRightRotate` function takes a node pointer `t` and returns a pointer to its right child's left child. It handles cases where `t`'s left child is not NULL. The `singleLeftRotate` function is similar, handling cases where `t`'s right child is not NULL.
- Compiler Output:** Shows compilation results with 0 errors and 0 warnings. The output file is "D:\dev\21\_april\q1\Project1.exe", size is 2.20517635345459 MiB, and compilation time is 1.78s.

**Bottom Window (Right):**

- Project Explorer:** Shows the same project structure.
- Code Editor:** Displays the implementation of an AVL tree search function. The function `T AVL<T>::find(const T& x, typename AVL<T>::node\* t)` traverses the tree based on the comparison of the search value `x` with the current node's data. It returns 0 if the node is NULL, 1 if the node is found, and 0 if the node is not found.
- Compiler Output:** Shows compilation results with 0 errors and 0 warnings. The output file is "D:\dev\21\_april\q1\Project1.exe", size is 2.20517635345459 MiB, and compilation time is 1.78s.

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

68     else if(x > t->data)
69     {
70         t->right = insert(x, t->right);
71         if(height(t->right) - height(t->left) == 2)
72         {
73             if(x > t->right->data)
74                 | t = singleLeftRotate(t);
75             else
76                 | t = doubleLeftRotate(t);
77         }
78     }
79
80     t->height = max(height(t->left), height(t->right))+1;
81     return t;
82 }

83
84
85 template <typename T>
86 T AVL<T>::find(const T& x, typename AVL<T>::node* t)
87 {
88
89     if(t == NULL)
90     {
91         return 0;
92     }

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

46     template <typename T>
47     typename AVL<T>::node* AVL<T>::insert(const T& x, typename AVL<T>::node* t)
48     {
49
50         if(t == NULL)
51         {
52             t = new typename AVL<T>::node;
53             t->data = x;
54             t->height = 0;
55             t->left = t->right = NULL;
56         }
57         else if(x < t->data)
58         {
59             t->left = insert(x, t->left);
60             if(height(t->left) - height(t->right) == 2)
61             {
62                 if(x < t->left->data)
63                     | t = singleRightRotate(t);
64                 else
65                     | t = doubleRightRotate(t);
66             }
67         }
68         else if(x > t->data)
69         {

```

Compiler Resources CompileLog Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

24
25     template <typename T>
26     void AVL<T>:: postorder(typename AVL<T>::node* p)
27     {
28         if (p!=NULL)
29         {
30             postorder(p->left);
31             postorder(p->right);
32             cout<<p->data<<" ";
33         }
34     }
35
36     template <typename T>
37     void AVL<T>:: makeEmpty(typename AVL<T>::node* t)
38     {
39         if(t == NULL)
40             return;
41         makeEmpty(t->left);
42         makeEmpty(t->right);
43         t=NULL;
44         delete t;
45     }
46
47     template <typename T>

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

1 #include "AVL.h"
2
3 #include<vector>
4 #include<iostream>
5 using namespace std;
6
7 template class AVL<char>;
8 template class AVL<int>;
9
10
11
12
13 template <typename T>
14 void AVL<T>::preorder(typename AVL<T>::node* p)
15 {
16     if (p!=NULL)
17     {
18         cout<<p->data<<" ";
19         preorder(p->left);
20         preorder(p->right);
21     }
22 }
23
24

```

Compiler Resources CompileLog ✓ Debug Find Results (27) Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 327 Col: 5 Sel: 0 Lines: 450 Length: 9026 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

70     int height(node*t);
71
72     T getBalance(node* t);
73
74     void inorder(node* t);
75
76
77     AVL();
78
79     void insert(T x);
80
81     void remove(T x);
82
83
84     void display();
85
86
87     T getdata(node *t);
88
89
90 };
91
92 #endif
93

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

64     node* findMax(node* t);
65
66     node* remove(T x, node* t);
67
68     int height(node*t);
69
70     T getBalance(node* t);
71
72     void inorder(node* t);
73
74
75     AVL();
76
77     void insert(T x);
78
79     void remove(T x);
80
81
82     void display();
83
84
85
86
87
88
89
90
91
92
93

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds

Type here to search

18:36 22-04-2021 ENG

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

50
51     node* singleRightRotate(node* &t);
52
53     node* singleLeftRotate(node* &t);
54
55
56     node* doubleLeftRotate(node* &t);
57
58     node* doubleRightRotate(node* &t);
59
60     node* findMin(node* t);
61
62     node* findMax(node* t);
63
64     node* remove(T x, node* t);
65
66     int height(node*t);
67
68     T getBalance(node* t);

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds

Type here to search

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

37 void postorder(node* p);
38
39     void makeEmpty(node* t);
40
41     node* insert(const T& x, node* t);
42
43
44     T find(const T& x, node* t);
45
46     node* singleRightRotate(node* &t);
47
48     node* singleLeftRotate(node* &t);
49
50
51     node* doubleLeftRotate(node* &t);
52
53     node* doubleRightRotate(node* &t);
54
55
56     node* remove(T x, node* t);
57
58     int height(node*t);
59
60     T getBalance(node* t);

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds

Type here to search

18:36 22-04-2021 ENG

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

23     node* root;
24
25     void preorder(node* p);
26
27     vector<T> longestPath(node* root);
28     vector<T> parent(T val, node* root);
29
30     vector<T> child(T val, node* root);
31
32     node* addressofval(T val, node* t);
33
34     void postorder(node* p);
35
36     void makeEmpty(node* t);
37
38     node* insert(const T& x, node* t);
39
40     node* minValue(node* t);
41
42     void printInorder(node* t);
43
44     void printPreorder(node* t);
45
46

```

Compiler Resources Compile Log ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds 18:36 22-04-2021

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

1 #ifndef AVL_H
2 #define AVL_H
3
4
5 #include<iostream>
6 #include<vector>
7 using namespace std;
8
9
10 template <typename T>
11 class AVL
12 {
13 public:
14     struct node
15     {
16         T data="#";
17         node* left;
18         node* right;
19         int height=-1;
20     };
21
22     node* root;
23
24

```

Compiler Resources Compile Log ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 18 Col: 18 Sel: 0 Lines: 93 Length: 1202 Insert Done parsing in 0.031 seconds 18:36 22-04-2021

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

137 cout << "\n\n Enter VALUE to find in AVL Tree: ";
138 cin >> val;
139
140 if(at.find(val,at.root))
141 {
142     cout<< "\n\n    present ";
143 }
144 else{
145     cout<< "\n\n    absent ";
146 }
147 break;
148
149
150
151
152 default:
153     cout << "\n\n    Enter Proper Option number " << endl;
154 }
155
156 } while (option != 0);
157
158 return 0;
159 }
```

Compiler Resources Compile Log ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

133 case 7:
134
135     cout << "\n\n Enter VALUE to find in AVL Tree: ";
136     cin >> val;
137
138 if(at.find(val,at.root))
139 {
140     cout<< "\n\n    present ";
141 }
142 else{
143     cout<< "\n\n    absent ";
144 }
145 break;
146
147
148
149
150
151
152 default:
153     cout << "\n\n    Enter Proper Option number " << endl;
154 }
155
156 } while (option != 0);
157
158 return 0;
159 }
```

Compiler Resources Compile Log ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

16:36 22-04-2021 ENG

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

109
110     case 6:
111
112         cout << "\n\n      maximum ";
113         val=at.getdata(at.findMax(at.root));
114
115         cout<<val;
116         output = at.parent(val,at.root);
117         n= output.size();
118         if(n==0)
119         {
120             | cout<< "\n\n      root has no parent or grandparent ";
121
122
123         else if(n==1)
124         {
125             | cout<<"\n\n      no grandparent";
126         }
127         else
128             cout<<"\n\n      grandparent : "<<output[n-2];
129
130
131         break;
132

```

Compiler Resources Compile Log ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

99
100    cout<<"\n\n      parent has no sibling ";
101
102    else
103    {
104        cout<<"\n\n      parent's sibling' : "<<aunt;
105
106
107
108
109
110
111
112     case 6:
113
114         cout << "\n\n      maximum ";
115         val=at.getdata(at.findMax(at.root));
116
117         cout<<val;
118         output = at.parent(val,at.root);
119         n= output.size();
120         if(n==0)
121         {
122             | cout<< "\n\n      root has no parent or grandparent ";
123
124
125
126
127
128
129
130
131
132

```

Compiler Resources Compile Log ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

16:36 22-04-2021 ENG

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

81
82
83
84     case 5:
85
86         cout << "\n\n      minimum    "; val=at.getdata(at.findMin(at.root));
87
88         cout<<val;
89
90         output = at.parent(val,at.root);
91         n= output.size();
92
93         chi = at.child(val,at.root);
94         if(chi[n-1]==output[n-1])
95             | aunt=chi[n-1];
96         else aunt=chi[n-2];
97
98         if(aunt=='#')
99         {
100             cout<<"\n\n      parent has no sibling ";
101         }
102         else
103         {
104             cout<<"\n\n      parent's sibling' : "<<aunt;

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

60
61         at.remove(val);
62         break;
63     case 3:
64         at.display();
65         break;
66
67     case 4:
68
69         cout << "\n\n height of tree      ";
70
71         cout<<at.height(at.root);
72         output = at.longestPath(at.root);
73         n= output.size();
74         cout<<"\n\n longest path:\n\n      ";
75         cout << output[n-1];
76         for (int i = n - 2; i >= 0; i--) {
77             cout << " -> " << output[i];
78         }
79         break;
80
81
82     case 5:
83

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Type here to search

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

46 {
47     | cout<< "\n\n      root has no parent or grandparent ";
48 }
49
50 else if(n==1)
51 {
52     | cout<<"\n\n      no grand parent";cout<< "\n\n      parent :   "<<output[n-1];
53 }
54 else
55     cout<<"\n\n      parent :   "<<output[n-1]<<"\n\n      grandparent :   "<<output[n-2];
56
57
58
59
60 at.remove(val);
61 break;
62 case 3:
63     at.display();
64 break;
65
66 case 4:
67
68     cout << "\n\n height of tree      ";
69

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```

28 switch (option) {
29 {
30     case 0:
31         break;
32     case 1:
33         cout << "\n\n      AVL INSERT" << endl;
34         cout << "\n\n      Enter VALUE of TREE NODE to INSERT in AVL Tree:   ";
35         cin >> val;
36         at.insert(val);
37         break;
38
39     case 2:
40         cout << "\n\n      DELETE" << endl;
41         cout << "\n\n      Enter VALUE of TREE NODE to DELETE in AVL:   ";
42         cin >> val;
43         output = at.parent(val,at.root);
44         n= output.size();
45         if(n==0)
46         {
47             | cout<< "\n\n      root has no parent or grandparent ";
48         }
49
50     else if(n==1)
51

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.78s

Line: 26 Col: 5 Sel: 0 Lines: 160 Length: 3580 Insert Done parsing in 0.031 seconds

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp AVL.h AVL.cpp

```
1 #include "AVL.h"
2 #include<iostream>
3 #include<vector>
4 using namespace std;
5
6 int main() {
7     AVL<char> at;
8     int option,int n;
9     char val,int hleft=-1,hright=-1;
10    vector<char> output;
11    vector<char> chij,char aunt;
12
13    do {
14        cout << "\n\n\n What operation do you want to perform? Select Option number" << endl;
15        cout << "\n      1. Insert Node" << endl;
16        cout << "\n      2. Delete Node and return its parent and grandparent" << endl;
17        cout << "\n      3. display" << endl;
18        cout << "\n      4. Height of Tree and display that longest path path" << endl;
19        cout << "\n      5. find min of Tree and it's parent's sibling'" << endl;
20        cout << "\n      6. find max of Tree and its grandpa" << endl;
21        cout << "\n      7. find Node" << endl;
22        cout << "\n      8. Exit Program\n\n      ";
23    }
```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

-----

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\ql\Project1.exe
- Output Size: 2.20517635345459 MiB
- Compilation Time: 1.70s

Type here to search

Line: 25 Col: 19 Sel: 0 Lines: 160 Length: 3590 Insert Done parsing in 0.031 seconds

18:35 22-04-2021 ENG

D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

7

Enter VALUE to find in AVL Tree: y

absent

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

0

---

Process exited after 104.3 seconds with return value 0



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

7

Enter VALUE to find in AVL Tree: h

present

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

7



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

6

maximum h

grandparent : f

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

7

Enter VALUE to find in AVL Tree: h



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

5

minimum a

parent has no sibling

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

6

maximum h



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

4

height of tree 3

longest path:

d -> f -> g -> h

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

5



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

3

inorder traversal :  
a b c d e f g h

preorder traversal :  
d b a c f e g h

postorder traversal :  
a c b e h g f d

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: h

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

3



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: g

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: f

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

3

inorder traversal :

a b c d e

preorder traversal :

d b a c e

postorder traversal :

a c b e d

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

2

DELETE

Enter VALUE of TREE NODE to DELETE in AVL: f

no grand parent

parent : d

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

2

DELETE

Enter VALUE of TREE NODE to DELETE in AVL: g

parent : f  
grandparent : d

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program



Type here to search



D:\dev\21 april\q1\Project1.exe

## AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: h

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

2

## DELETE

Enter VALUE of TREE NODE to DELETE in AVL: h

parent : g

grandparent : f

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: h

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

2



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: g

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: f

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: e

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: d

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21.april\q1\Project1.exe

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: b

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: c

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node



Type here to search



D:\dev\21 april\q1\Project1.exe

Enter VALUE of TREE NODE to INSERT in AVL Tree: a

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: b

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program



Type here to search



D:\dev\21 april\q1\Project1.exe

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1

AVL INSERT

Enter VALUE of TREE NODE to INSERT in AVL Tree: a

What operation do you want to perform? Select Option number

1. Insert Node
2. Delete Node and return its parent and grandparent
3. display
4. Height of Tree and display that longest path path
5. find min of Tree and it's parent's sibling''
6. find max of Tree and its grandpa
7. find Node
0. Exit Program

1



Type here to search



D:\dev\21 april\q1\Project1.exe

- 6. find max of Tree and its grandpa
- 7. find Node
- 0. Exit Program

7

Enter VALUE to find in AVL Tree: y

absent

What operation do you want to perform? Select Option number

- 1. Insert Node
- 2. Delete Node and return its parent and grandparent
- 3. display
- 4. Height of Tree and display that longest path path
- 5. find min of Tree and it's parent's sibling''
- 6. find max of Tree and its grandpa
- 7. find Node
- 0. Exit Program

0

-----  
Process exited after 104.3 seconds with return value 0  
Press any key to continue . . .



Type here to search



2.Create a Telephone dictionary which consists of Telephone number, Name and Address of the customer (Use AVL tree for implementation). Perform the following operations:

- a. Insert into the dictionary whenever new customer comes.
- b. Delete the entry from the dictionary if the customer leaves.
- c. If name is given as input, corresponding telephone number and address needs to be retrieved.
- d. Change of address can be provided to the customers (But change of customer name and telephone number should not be provided. Use telephone number as key)
- e. Suitable display operations for every operation stated above.
- f. Display of names in the dictionary if partial match in the name occurs (partial match occurs for 3 letters – Assume name can be a maximum of 7 characters)

## Main.cpp

```
#include<cstring>
#include<iostream>
#include<cstdlib>
#include<cstdio>
#include "phone.h"

using namespace std;

int main()
{
    char name[100] ,address[100];
    int contact;
    int num;
    node* head=NULL;
    int ch , rtrn;
    AVL avl;
```

```

do
{
    cout<<"\n\n\n\n\n\n phone directory"<<endl;

    cout<<"\n\n    1. insert new contact"
        <<"\n\n    2. display directory"
        <<"\n\n    3. seach contact by name"
        <<"\n\n    4.delete contact by name"
        <<"\n\n    5. update adress of contact with number as key"
        <<"\n\n    6.display partially matched names(3 chars)\n"
        <<"\n\n    0. exit"

<<"\n\n\n    enter choice:   ";
    cout<<"  ";cin>>ch;
    switch(ch)
    {
        case 1: cout<<"\n\n\n\n      Enter Name(atleast 3 characters) :  ";
                    cin>>name;
                    cout<<"\n\n      Enter phone number :  ";
                    cin>>contact;
                    cout<<"\n\n      Enter Address :  ";
                    cin>>address;
                    head = avl.insert( name , contact,address, head);
                    break;

        case 2: if(head!=NULL)
        {

```

```
cout<<"\n\n      Contacts Present in the  
Directory are:\n\n    "<<endl;
```

```
        avl.inorder(head);
```

```
        cout<<"\n\n\n";
```

```
}
```

```
else
```

```
    cout<<"\n directory is empty"<<endl;
```

```
break;
```

```
case 3: cout<<"\n\n\n      Enter Name to Find Contact\n\n"<<endl;
```

```
        cout<<"      ";cin>>name;
```

```
        rtrn =avl.search(head , name);
```

```
        if(rtrn!=1)
```

```
{
```

```
        cout<<"contact is not present"<<endl;
```

```
}
```

```
break;
```

```
case 4: cout<<"\n\n\n      Enter Contact to be Deleted\n\n"<<endl;
```

```
        cout<<"      ";cin>>name;
```

```
        rtrn =avl.search(head , name);
```

```
        if(head==NULL)
```

```
{
```

```
        cout<<"\n      directory is empty"<<endl;
```

```
}
```

```
        else if(head!=NULL && rtrn ==1)
```

```
{
```

```
        head= avl.delete_node(head , name);
```

```

cout<<"\n\n    deleted sucessfully";
}

else

    cout<<"\ncontact is not present"<<endl;

break;

case 5: cout<<"\n\n\n    Enter phone number to Update the address
\n\n"<<endl;

cout<<"      ";cin>>num;

avl.update(head , num);

break;

case 0: break;

case 6:

    cout<<"\n\n\n    Enter partial name to be found( atleast 3 chars
should match ) \n\n";

    cout<<"      ";cin>>name;

    cout<<"\n\n  ";

    avl.searchpartial(head , name);

break;

default: cout<<"\ninvalid choice"<<endl;

}

}while(ch!=0);

return 0;
}

```

## Phone.h

```
#ifndef PHONE_H
#define PHONE_H

class node
{
public:
    char name[128] ,address[256];
    int contact;
    class node *left;
    class node *right;
    int height;
};

class AVL
{
public:
    node* root
    ;
    node*& insert(char* , int ,char*, node*&);

    void inorder(node*&);

    int search(node*& , char*);

    node*& min(node*&);

    node*& delete_node(node*& , char*);

    void update(node*& , int);
```

```
int height(node*&);

int BF(node*&);

node*& left_right(node*&);

node*& left_left(node*&);

node*& right_left(node*&);

node*& right_right(node*&);

node* left_rotate(node*&);

node* right_rotate(node*&)

void searchpartial(node*&, char *);

tree()
{
    root=NULL;
}

};

#endif
```

## Phone.cpp

```
#include<cstring>
#include<iostream>
#include<cstdlib>
#include<cstdio>
#include "phone.h"

using namespace std;

#include "phone.h"

int AVL :: height(node*& T)
{
    int lh , rh;
    if(T->left==NULL)
        lh=0;
    else
        lh=1+T->left->height;
    if(T->right==NULL)
        rh=0;
    else
        rh=1+T->right->height;
    if(lh>rh)
        return lh;
    return rh;
}
```

```

int AVL :: BF(node*& T)
{
    int lh , rh;
    if(T==NULL)
        return 0;
    if(T->left==NULL)
        lh=0;
    else
        lh=1+T->left->height;
    if(T->right==NULL)
        rh=0;
    else
        rh=1+T->right->height;
    return (lh-rh);
}

```

```

node*& AVL :: right_right(node *&T)
{
    T=left_rotate(T);
    return T;
}

node*& AVL :: left_left(node *&T)
{
    T=right_rotate(T);
    return T;
}

```

```

node*& AVL :: left_right(node*& T)

```

```
{  
    T->left=left_rotate(T->left);  
    T=right_rotate(T);  
    return T;  
}
```

```
node*& AVL :: right_left(node*& T)  
{  
    T->right=right_rotate(T->right);  
    T=left_rotate(T);  
    return T;  
}
```

```
node* AVL :: right_rotate(node*& x)  
{  
    node *y ;  
    y=x->left;  
    x->left=y->right;  
    y->right=x;  
    x->height=height(x);  
    y->height=height(y);  
    return y;  
}
```

```
node* AVL :: left_rotate(node*& x)  
{  
    node *y ;  
    y=x->right;  
    x->right=y->left;
```

```

y->left=x;
x->height=height(x);
y->height=height(y);
return y;
}

node*& AVL :: min(node *&q)
{
    while(q->left != NULL)
    {
        q = q->left;
    }
    return q;
}

node*& AVL :: insert(char* name , int contact,char* address , node*& T)
{
    int cmp;
    if(T==NULL)
    {

        T=new node;
        T->left=NULL;
        T->right=NULL;
        strcpy(T->name , name);
        strcpy(T->address , address);
        T->contact=contact;
    }
}

```

```

else
{
    cmp=strcasecmp( name , T->name);
    if(cmp<0)
    {
        T->left=insert(name , contact ,address,T->left);
        if(BF(T)==2)
        {
            if(strcasecmp( name , T->left->name)<0)
                T=left_left(T);
            else
                T=left_right(T);
        }
    }

    else if(cmp>0)
    {
        T->right =insert(name ,contact ,address ,T->right);
        if(BF(T)==-2)
        {
            if(strcasecmp( name , T->right->name)>0)
                T=right_right(T);
            else
                T=right_left(T);
        }
    }

    else
    {
}
}

```

```

        cout<<"contact already exists"<<endl;
    }

}

T->height=height(T);

return T;

}

void AVL ::inorder(node*& head)
{
    if(head!=NULL)
    {
        inorder(head->left);

        cout<<"\n\n  Name : "<<head->name<<"      phone number : "
"<<head->contact<<"      address : "<<head->address<<"\n\n";
        inorder(head->right);
    }
}

int AVL :: search(node*& head , char *str)
{
    while(head!=NULL)
    {
        int cmp=strcasecmp( str , head->name);

        if(cmp==0)
        {
            cout<<"\n\n\n  found record \n\n";
            cout<<"\n\n  name is :\t"<<head->name<<endl;
        }
    }
}

```

```

        cout<<"\n\n  contact is :\t"<<head->contact<<endl;
        cout<<"\n\n  address is :\t"<<head->address<<endl;

    return 1;

}

else if(cmp<0)

{

    head=head->left;

}

else if(cmp>0)

{

    head=head->right;

}

}

return -1;

}

void AVL :: searchpartial(node*& head , char *str)
{
    if(head==NULL) return;
    else
    {int i=0,count=0;
        while(str[i]!='\0' && (head->name)[i]!='\0' )
    {

```

```

        if(str[i]==(head->name)[i])
        {
            count++;
        }
        i++;
    }

    if(count>=3)
    {
        cout<<"\n\n      "<<head->name<<"\n";
    }

    searchpartial(head->right,str);
    searchpartial(head->left,str);
}

}

node*& AVL :: delete_node(node*& T , char* str )
{
    node *temp;
    int cmp;
    cmp=strcasecmp(str , T->name);
    if(T==NULL)
    {
        cout<<"Phone Directory not created"<<endl;
        return T;
    }
}

```

```

else if(cmp<0)

{
    T->left=delete_node(T->left , str);
    if(BF(T)==-2)
    {
        if(BF(T->right)<=0)
            T=right_right(T);
        else
            T=right_left(T);
    }
}

else if(cmp>0)

{
    T->right=delete_node(T->right , str);
    if(BF(T)==2)
    {
        if(BF(T->left)>=0)
            T=left_left(T);
        else
            T=left_right(T);
    }
}

else
{
    if(T->right!=NULL)
    {
        temp=T->right;

```

```

        while(temp->left!=NULL)
            temp=temp->left;
        strcpy(T->name , temp->name);
        T->contact=temp->contact;
        strcpy(T->address , temp->address);
        T->right=delete_node(T->right , temp->name);
        if(BF(T)==2)
        {
            if(BF(T->left)>=0)
                T=left_left(T);
            else
                T=left_right(T);
        }
    }
    else
    {
        return T->left;
        cout<<"\nSorry entered element not found\n"<<endl;
    }
}
T->height=height(T);
return T;
}

void AVL :: update(node*& head , int contact)
{

```

```
char new_address[256];

if(head==NULL)
{
    return ;
}

else{

    if(contact==head->contact)
    {
        cout<<"\n\n\n      Enter the new address   ";
        cout<<"      ";cin>>new_address;
        strcpy(head->address , new_address);
        cout<<"\n\n      updated sucessfully\n\n";

    }

    update(head->right,contact);
    update(head->left,contact);

}

}
```

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

79
80
81     case 5: cout<<"\n\n\n      Enter phone number to Update the address      \n\n"endl;
82     cout<<"      ";cin>>num;
83     avl.update(head , num);
84
85     break;
86
87     case 0: break;
88
89     case 6:
90         cout<<"\n\n\n      Enter partial name to be found( atleast 3 chars should match ) \n\n";
91         cout<<"      ";cin>>name;
92         cout<<"\n\n      ";
93         avl.searchpartial(head , name);
94
95         break;
96
97     default: cout<<"\ninvalid choice"endl;
98
99 }
100
101 while(ch!=0);
102 return 0;

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Line: 14 Col: 10 Sel: 0 Lines: 102 Length: 2529 Insert Done parsing in 0.047 seconds 22:42 22-04-2021

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

77
78     else
79         cout<<"\ncontact is not present"endl;
80         break;
81
82     case 5: cout<<"\n\n\n      Enter phone number to Update the address      \n\n"endl;
83     cout<<"      ";cin>>num;
84     avl.update(head , num);
85
86     break;
87
88     case 0: break;
89
90     case 6:
91         cout<<"\n\n\n      Enter partial name to be found( atleast 3 chars should match ) \n\n";
92         cout<<"      ";cin>>name;
93         cout<<"\n\n      ";
94         avl.searchpartial(head , name);
95
96         break;
97
98     default: cout<<"\ninvalid choice"endl;
99
100
101 while(ch!=0);

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Line: 14 Col: 10 Sel: 0 Lines: 102 Length: 2529 Insert Done parsing in 0.047 seconds 22:42 22-04-2021

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

53     |         break;
54
55     case 3: cout<<"\n\n\n      Enter Name to Find Contact\n\n" << endl;
56             cout<<"      "; cin>>name;
57             rtrn = avl.search(head , name);
58             if(rtrn!=1)
59             {
60                 cout<<"contact is not present" << endl;
61             }
62
63             break;
64
65     case 4: cout<<"\n\n\n      Enter Contact to be Deleted\n\n" << endl;
66             cout<<"      "; cin>>name;
67             rtrn = avl.search(head , name);
68             if(head==NULL)
69             {
70                 cout<<"\n      directory is empty" << endl;
71             }
72             else if(head!=NULL && rtrn ==1)
73             {
74                 head= avl.delete_node(head , name);
75                 cout<<"\n\n      deleted sucessfully";
76             }

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 14 Col: 10 Sel: 0 Lines: 102 Length: 2529 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

35             cin>>name;
36             cout<<"\n\n      Enter phone number : ";
37             cin>>contact;
38             cout<<"\n\n      Enter Address : ";
39             cin>>address;
40             head = avl.insert( name , contact,address, head);
41             break;
42
43     case 2: if(head!=NULL)
44     {
45
46         cout<<"\n\n      Contacts Present in the Directory are:\n\n" << endl;
47         avl.inorder(head);
48         cout<<"\n\n\n";
49     }
50     else
51         cout<<"\n      directory is empty" << endl;
52         break;
53
54     case 3: cout<<"\n\n\n      Enter Name to Find Contact\n\n" << endl;
55             cout<<"      "; cin>>name;
56             rtrn = avl.search(head , name);
57             if(rtrn!=1)
58

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 14 Col: 10 Sel: 0 Lines: 102 Length: 2529 Insert Done parsing in 0.047 seconds



Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

17 {
18     public:
19     node* root;
20     node*& insert(char*, int, char*, node*&);
21     void inorder(node*&);

22     int search(node*&, char*);
23     node*& min(node*&);
24     node*& delete_node(node*&, char*);
25     void update(node*&, int);
26     int height(node*&);

27     int BF(node*&);

28     node*& left_right(node*&);

29     node*& left_left(node*&);

30     node*& right_left(node*&);

31     node*& right_right(node*&);

32     node*& left_rotate(node*&);

33     node*& right_rotate(node*&);

34     void searchpartial(node*&, char *);

35     void tree();
36     {
37         root=NULL;
38     }
39 }
40 
```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 34 Col: 10 Sel: 0 Lines: 41 Length: 695 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h phone.cpp

```

1
2
3
4 #ifndef PHONE_H
5 #define PHONE_H
6 class node
7 {
8     public:
9     char name[128], address[256];
10    int contact;
11    class node *left;
12    class node *right;
13    int height;
14 };
15
16 class AVL
17 {
18     public:
19     node* root;
20     node*& insert(char*, int, char*, node*&);
21     void inorder(node*&);

22     int search(node*&, char*);
23     node*& min(node*&);

24 
```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 34 Col: 10 Sel: 0 Lines: 41 Length: 695 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

293 }
294 else{
295
296
297
298
299
300 if(contact==head->contact)
301 {
302     cout<<"\n\n\n      Enter the new address      ";
303     cout<<"\n      ";cin>>new_address;
304     strcpy(head->address , new_address);
305     cout<<"\n\n      updated sucessfully\n\n";
306
307
308
309     update(head->right,contact);
310     update(head->left,contact);
311
312
313 }
314
315 }
```

Compiler Resources CompileLog ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

288 {
289     char new_address[256];
290     if(head==NULL)
291     {
292         return ;
293     }
294     else{
295
296
297
298
299
300     if(contact==head->contact)
301     {
302         cout<<"\n\n\n      Enter the new address      ";
303         cout<<"\n      ";cin>>new_address;
304         strcpy(head->address , new_address);
305         cout<<"\n\n      updated sucessfully\n\n";
306
307
308
309     update(head->right,contact);
310     update(head->left,contact);
311
312
313 }
```

Compiler Resources CompileLog ✓ Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project2 Classes Debug main.cpp phone.h phone.cpp

```
273 |     }
274 | }
275 |
276 | {
277 |     T->height=height(T);
278 |     return T;
279 |
280 | }
281 |
282 | T->height=height(T);
283 | return T;
284 |
285 | }
286 |
287 void AVL :: update(node*& head , int contact)
288 {
289     char new_address[256];
290     if(head==NULL)
291     {
292         return ;
293     }
294     else{
295 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project2 Classes Debug main.cpp phone.h phone.cpp

```
251 |     else
252 |         T=left_right(T);
253 |
254 |
255 |
256 |     else
257 |     {
258 |         if(T->right!=NULL)
259 |         {
260 |             temp=T->right;
261 |             while(temp->left!=NULL)
262 |                 temp=temp->left;
263 |             strcpy(T->name , temp->name);
264 |             T->contact=temp->contact;
265 |             strcpy(T->address , temp->address);
266 |             T->right=delete_node(T->right , temp->name);
267 |             if(BF(T)==2)
268 |             {
269 |                 if(BF(T->left)>0)
270 |                     T=left_left(T);
271 |                 else
272 |                     T=left_right(T);
273 |             }
274 |         }
275 |     }
276 | }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

232     else if(cmp<0)
233     {
234         T->left=delete_node(T->left , str);
235         if(BF(T)==-2)
236         {
237             if(BF(T->right)<=0)
238                 T=right_right(T);
239             else
240                 T=right_left(T);
241         }
242     }
243
244     else if(cmp>0)
245     {
246         T->right=delete_node(T->right , str);
247         if(BF(T)==2)
248         {
249             if(BF(T->left)>=0)
250                 T=left_left(T);
251             else
252                 T=left_right(T);
253         }
254     }

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

213     }
214     searchpartial(head->right,str);
215     searchpartial(head->left,str);
216 }
217
218
219
220
221 node*& AVL :: delete_node(node*& T , char* str )
222 {
223     node *temp;
224     int cmp;
225     cmp=strcasecmp(str , T->name);
226     if(T==NULL)
227     {
228         cout<<"Phone Directory not created"<<endl;
229         return T;
230     }
231
232     else if(cmp<0)
233     {
234         T->left=delete_node(T->left , str);
235         if(BF(T)==-2)
236         {
237             if(BF(T->right)<=0)
238                 T=right_right(T);
239             else
240                 T=right_left(T);
241         }
242     }
243
244     else if(cmp>0)
245     {
246         T->right=delete_node(T->right , str);
247         if(BF(T)==2)
248         {
249             if(BF(T->left)>=0)
250                 T=left_left(T);
251             else
252                 T=left_right(T);
253         }
254     }

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

192
193
194 void AVL :: searchpartial(node*& head , char *str)
195 {
196
197     if(head==NULL) return;
198     else
199     {
200         int i=0,count=0;
201         while(str[i]!='\0' && (head->name)[i]!='\0' )
202         {
203             if(str[i]==(head->name)[i])
204             {
205                 count++;
206             }
207             i++;
208
209             if(count==3)
210             {
211                 cout<<"\n\n"           <<head->name<<"\n";
212             }
213         }
214         searchpartial(head->right,str);

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

178
179
180     }
181     else if(cmp<0)
182     {
183         head=head->left;
184     }
185     else if(cmp>0)
186     {
187         head=head->right;
188     }
189
190     return -1;
191 }
192
193
194 void AVL :: searchpartial(node*& head , char *str)
195 {
196
197     if(head==NULL) return;
198     else
199     {
200         int i=0,count=0;
201         while(str[i]!='\0' && (head->name)[i]!='\0' )

```

Compiler Resources CompileLog ✓ Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

161 }
162 }
163 int AVL :: search(node*& head , char *str)
164 {
165     while(head!=NULL)
166     {
167         int cmp=strcasecmp( str , head->name);
168         if(cmp==0)
169         {
170             cout<<"\n\n\n    found record  \n\n";
171             cout<<"\n\n        name is :\t"<<head->name<<endl;
172             cout<<"\n\n        contact is :\t"<<head->contact<<endl;
173             cout<<"\n\n        address is :\t"<<head->address<<endl;
174
175             return 1;
176         }
177         else if(cmp<0)
178         {
179             head=head->left;
180         }
181         else if(cmp>0)
182     }
183 }
```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

151
152 void AVL ::inorder(node*& head)
153 {
154     if(head!=NULL)
155     {
156         inorder(head->left);
157         cout<<"\n\n      Name : "<<head->name<<"          phone number : "<<head->contact<<"          address : "<<head->address<<
158         inorder(head->right);
159     }
160
161 }
162
163 int AVL :: search(node*& head , char *str)
164 {
165     while(head!=NULL)
166     {
167         int cmp=strcasecmp( str , head->name);
168         if(cmp==0)
169         {
170             cout<<"\n\n\n    found record  \n\n";
171             cout<<"\n\n        name is :\t"<<head->name<<endl;
172             cout<<"\n\n        contact is :\t"<<head->contact<<endl;
173 }
```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds



Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

91 node*& AVL :: min(node *&q)
92 {
93     while(q->left != NULL)
94     {
95         q = q->left;
96     }
97     return q;
98 }
99
100 node*& AVL :: insert(char* name , int contact,char* address , node*& T)
101 {
102     int cmp;
103     if(T==NULL)
104     {
105         T=new node;
106         T->left=NULL;
107         T->right=NULL;
108         strcpy(T->name , name);
109         strcpy(T->address , address);
110         T->contact=contact;
111     }
112 }
113

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

70 node* AVL :: right_rotate(node*& x)
71 {
72     node *y ;
73     y=x->left;
74     x->left=y->right;
75     y->right=x;
76     x->height=height(x);
77     y->height=height(y);
78     return y;
79 }
80
81 node* AVL :: left_rotate(node*& x)
82 {
83     node *y ;
84     y=x->right;
85     x->right=y->left;
86     y->left=x;
87     x->height=height(x);
88     y->height=height(y);
89     return y;
90 }
91
92 node*& AVL :: min(node *&q)

```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

55
56     node*& AVL :: left_right(node*& T)
57 {
58     T->left=left_rotate(T->left);
59     T=right_rotate(T);
60     return T;
61 }
62
63     node*& AVL :: right_left(node*& T)
64 {
65     T->right=right_rotate(T->right);
66     T=left_rotate(T);
67     return T;
68 }
69
70     node*& AVL :: right_rotate(node*& x)
71 {
72     node *y ;
73     y=x->left;
74     x->left=y->right;
75     y->right=x;
76     x->height=height(x);
77     y->height=height(y);

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

44
45     node*& AVL :: right_right(node *&T)
46 {
47     T=left_rotate(T);
48     return T;
49 }
50     node*& AVL :: left_left(node *&T)
51 {
52     T=right_rotate(T);
53     return T;
54 }
55
56     node*& AVL :: left_right(node*& T)
57 {
58     T->left=left_rotate(T->left);
59     T=right_rotate(T);
60     return T;
61 }
62
63     node*& AVL :: right_left(node*& T)
64 {
65     T->right=right_rotate(T->right);
66     T=left_rotate(T);

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

28 int AVL :: BF(node*& T)
29 {
30     int lh , rh;
31     if(T==NULL)
32         return 0;
33     if(T->left==NULL)
34         lh=0;
35     else
36         lh=1+T->left->height;
37     if(T->right==NULL)
38         rh=0;
39     else
40         rh=1+T->right->height;
41     return (lh-rh);
42 }
43
44 node*& AVL :: right_right(node *&T)
45 {
46     T=left_rotate(T);
47     return T;
48 }
49
50 _ node*& AVL :: left_left(node *&T)

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

Project2 - [Project2.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp phone.h Phone.cpp

```

13 int AVL :: height(node*& T)
14 {
15     int lh , rh;
16     if(T->left==NULL)
17         lh=0;
18     else
19         lh=1+T->left->height;
20     if(T->right==NULL)
21         rh=0;
22     else
23         rh=1+T->right->height;
24     if(lh>rh)
25         return lh;
26     return rh;
27 }
28
29 int AVL :: BF(node*& T)
30 {
31     int lh , rh;
32     if(T==NULL)
33         return 0;
34     if(T->left==NULL)
35         lh=0;

```

Compiler Resources CompileLog Debug Find Results Close

About Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\dev\21\_april\q2\Project2.exe
- Output Size: 1.98921871185303 MiB
- Compilation Time: 1.63s

Type here to search

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

The screenshot shows the Dev-C++ IDE interface. The title bar reads "Project2 - [Project2.dev] - Dev-C++ 5.11". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help. The toolbar has icons for New, Open, Save, Build, Run, Stop, and others. The status bar at the bottom shows "Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds".

**Project2 - [Project2.dev] - Dev-C++ 5.11**

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug main.cpp phone.h phone.cpp

Project2

```
1 #include<cstring>
2 #include<iostream>
3 #include<cstdlib>
4 #include<cstdio>
5 #include "phone.h"
6
7 using namespace std;
8
9 #include "phone.h"
10
11
12
13 int AVL :: height(node*& T)
14 {
15     int lh , rh;
16     if(T->left==NULL)
17     | lh=0;
18     else
19     | lh=1+T->left->height;
20     if(T->right==NULL)
21     | rh=0;
22     else
23         rh=1+T->right->height;
```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation results...

-----

- Errors: 0  
- Warnings: 0  
- Output Filename: D:\dev\21\_april\q2\Project2.exe  
- Output Size: 1.90521871185303 MB  
- Compilation Time: 1.63s

Line: 218 Col: 2 Sel: 0 Lines: 315 Length: 4951 Insert Done parsing in 0.047 seconds

D:\dev\21 april\q2\Project2.exe

```
phone directory

1. insert new contact
2. display directory
3. seach contact by name
4.delete contact by name
5. update adress of contact with number as key
6.display partially matched names(3 chars)

0. exit
```

enter choice: 6

Enter partial name to be found( atleast 3 chars should match )

navin

navin

yatin

arvin

phone directory

1. insert new contact



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 6

Enter partial name to be found( atleast 3 chars should match )

xayin

navin

yatin

phone directory

1. insert new contact
2. display directory
3. seach contact by name



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 2

Contacts Present in the Directory are:

Name : arvin                      phone number : 9906                      address : chennai

Name : navin                      phone number : 7789                      address : mumbai

Name : yatin                      phone number : 76490                      address : russia



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  4. delete contact by name
  5. update adress of contact with number as key
  6. display partially matched names(3 chars)
0. exit

enter choice: 5

Enter phone number to Update the address

7789

Enter the new address mumbai

updated sucessfully

phone directory

1. insert new contact



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
2. display directory
3. seach contact by name
- 4.delete contact by name
5. update adress of contact with number as key
- 6.display partially matched names(3 chars)
0. exit

enter choice: 2

Contacts Present in the Directory are:

Name : arvin	phone number : 9906	address : chennai
Name : navin	phone number : 7789	address : coimbatore
Name : yatin	phone number : 76490	address : russia



Type here to search



D:\dev\21 april\q2\Project2.exe

```
phone directory

1. insert new contact
2. display directory
3. seach contact by name
4.delete contact by name
5. update adress of contact with number as key
6.display partially matched names(3 chars)

0. exit
```

enter choice: 4

Enter Contact to be Deleted

ashok

found record

name is : ashok

contact is : 8865

address is : trichy

deleted sucessfully



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 3

Enter Name to Find Contact

ashok

found record

name is : ashok

contact is : 8865

address is : trichy

phone directory



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 2

Contacts Present in the Directory are:

Name : arvin                      phone number : 9906                      address : chennai

Name : ashok                      phone number : 8865                      address : trichy

Name : navin                      phone number : 7789                      address : coimbatore

Name : yatin                      phone number : 76490                      address : russia



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
2. display directory
3. seach contact by name
- 4.delete contact by name
5. update adress of contact with number as key
- 6.display partially matched names(3 chars)
0. exit

enter choice: 1

Enter Name(atleast 3 characters) : ashok

Enter phone number : 8865

Enter Address : trichy

phone directory

1. insert new contact
2. display directory
3. seach contact by name
- 4.delete contact by name
5. update adress of contact with number as key



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 1

Enter Name(atleast 3 characters) : yatin

Enter phone number : 76490

Enter Address : russia

phone directory

1. insert new contact
2. display directory
3. seach contact by name



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
2. display directory
3. seach contact by name
- 4.delete contact by name
5. update adress of contact with number as key
- 6.display partially matched names(3 chars)
0. exit

enter choice: 1

Enter Name(atleast 3 characters) : arvin

Enter phone number : 9906

Enter Address : chennai

phone directory

1. insert new contact
2. display directory
3. seach contact by name



Type here to search



D:\dev\21 april\q2\Project2.exe

phone directory

1. insert new contact
2. display directory
3. seach contact by name
- 4.delete contact by name
5. update adress of contact with number as key
- 6.display partially matched names(3 chars)
0. exit

enter choice: 1

Enter Name(atleast 3 characters) : navin

Enter phone number : 7789

Enter Address : coimbatore

phone directory

1. insert new contact
2. display directory



Type here to search



D:\dev\21 april\q2\Project2.exe

navin

yatin

arvin

phone directory

1. insert new contact
  2. display directory
  3. seach contact by name
  - 4.delete contact by name
  5. update adress of contact with number as key
  - 6.display partially matched names(3 chars)
0. exit

enter choice: 0

-----  
Process exited after 209.9 seconds with return value 0  
Press any key to continue . . .