**CPP\_Assignment\_6b**

|  |  |  |
| --- | --- | --- |
| Write an application in C++ to read a line of text as a single string as command line argument, parse, extract, store and display the words following the specified steps: | | |
| Requirement Tag | Requirement Description | Comments |
| VSTR/01 | Read a line of text as a single string from command line |  |
| VSTR/02 | Create a class named “Wordset” with members as below vector <string> vlist;  int addmember(string s) - add given string s to vlist int deletemember(string s) – search and delete the given string s from vlist void displaylist() – iterate the vlist and display all its contents |  |
| VSTR/03 | Parse the command line arguments, extract words and store them in vlist |  |
| VSTR/04 | Prompt the user for a word to search and delete, then call deletemember if found |  |
| VSTR/05 | Display the contents of vector. |  |
| VSTR/06 | Empty the vlist and display the size of vlist |  |
| VSTR/07 | For every unque word stored in vector, get its frequency and display. | Hint: Use a map with key as string and frequency as value |

VSTR/01 Read a line of text as a single string from command line

Ans:

#include<iostream>

#include<vector>

#include<string>

using namespace std;

int main(int argc,char \*argv[])

{

int i;

for(i=0;i<argc;i++)

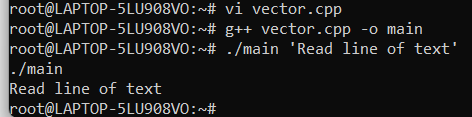
{

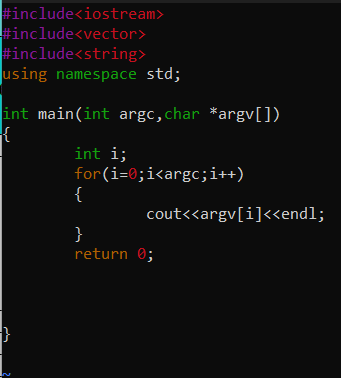
cout<<argv[i]<<endl;

}

return 0;

}





VSTR/02 Create a class named “Wordset” with members as below  
vector <string> vlist;  
  
int addmember(string s) - add given string s to vlist  
int deletemember(string s) – search and delete the given string s from vlist  
void displaylist() – iterate the vlist and display all its contents

Ans:

#include<iostream>

#include<vector>

#include<string>

using namespace std;

class Wordset{

private:

string s1="addmember";

string s2="deletemember";

public:

void add\_delete()

{

vector<string>vlist={"Read the file", "As single", "Command line"};

vlist.push\_back(s1);

vlist.push\_back(s2);

for(string s:vlist)

cout<<s<<" "<<endl;

vlist.pop\_back();

for(string s:vlist)

cout<<s<<" "<<endl;

vector<string>::iterator it;

for(it=vlist.begin();it<vlist.end();++it)

cout<<\*it<<" "<<endl;

}

};

int main(int argc,char \*argv[])

{

int i;

for(i=0;i<argc;i++)

{

cout<<argv[i]<<endl;

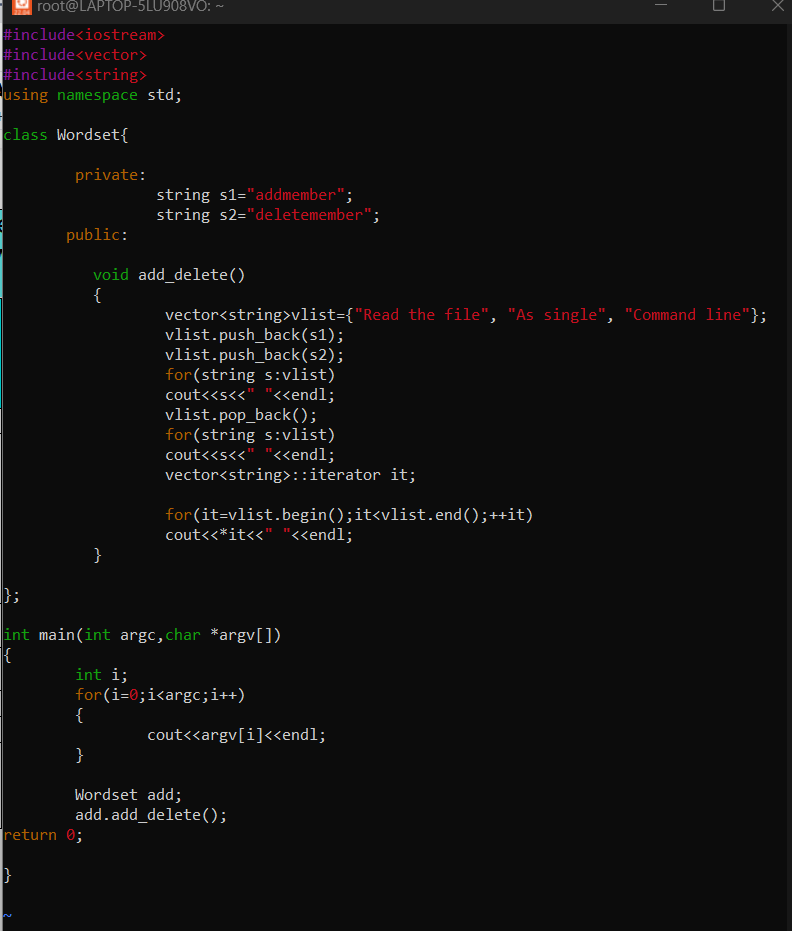
}

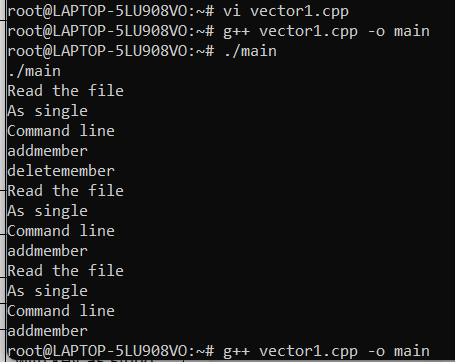
Wordset add;

add.add\_delete();

return 0;

}





VSTR/03 Parse the command line arguments, extract words and store them in vlist

VSTR/05 Display the contents of vector.

Ans:

#include<iostream>

#include<cstring>

#include<vector>

using namespace std;

int main(int argc,char \*argv[])

{ int i;

string str1,str2,str3,str4;

string str;

for(i=0;i<argc;i++)

{// cout<<argv[i]<<endl;

}

//str=argv[1];

//cout<<str<<endl;

str1=strtok(argv[1]," ");

//cout<<str1<<endl;

str2=strtok(argv[2]," ");

//cout<<str1<<endl;

str3=strtok(argv[3]," ");

//cout<<str1<<endl;

str4=strtok(argv[4]," ");

//cout<<str1<<endl;

vector<string>vlist={str1,str2,str3,str4};

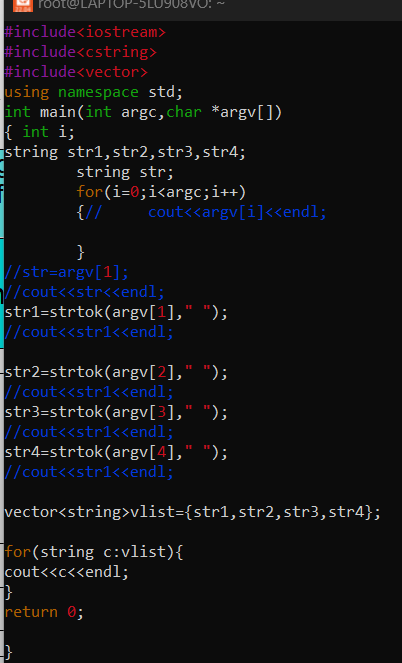
for(string c:vlist){

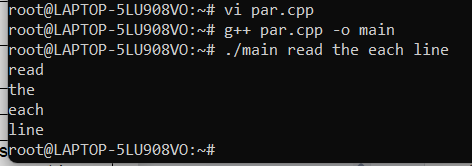
cout<<c<<endl;

}

return 0;

}





VSTR/04 Prompt the user for a word to search and delete, then call deletemember if found

Ans:

#include<iostream>

#include<vector>

#include<string>

using namespace std;

vector<string>vlist;

class Wordset{

private:

string s1="Durgadevi";

string s2="Rajanala";

public:

void addmember()

{

cout<<"Pushing data:";

vlist.push\_back(s1);

vlist.push\_back(s2);

cout<<" "<<endl;

}

void deletemember()

{

cout<<"\nPoping data:";

vlist.pop\_back();

cout<<" "<<endl;

}

void displaylist()

{

for(string s:vlist)

cout<<s<<" "<<endl;

}

};

int main(int argc,char \*argv[])

{

int i;

for(i=0;i<argc;i++)

{

cout<<"command line input is:";

cout<<argv[i]<<endl;

}

for(i=1;i<argc;i++)

{

vlist.push\_back(argv[i]);

}

cout<<endl;

cout<<"The vector is:"<<endl;

for(string s:vlist){

cout<<s<<"\n "<<endl;

}

Wordset add;

add.addmember();

add.displaylist();

add.deletemember();

add.displaylist();

cout<<"\nThe size of the vector before poping is:"<<vlist.size();

while(!vlist.empty())

{

vlist.pop\_back();

}

cout<<"\nThe size of the vector after poping is:"<<vlist.size()<<endl;;

return 0;

}

