CODE

#include <bits/stdc++.h>

using namespace std;

class string1

{

char str[80];

public:

void indata()

{

cout<<"Enter string";

gets(str);

}

void outdata()

{

cout<<str;

}

char\*retstr()

{

return str;

}

} ;

void WRITE()

{

ofstream f("secret.dat",ios::binary | ios::app);

string1 m;

char reply;

do

{

m.indata();

f.write((char\*)&m,sizeof(m));

cout<<"Want to enter more records";

cin>>reply;

}while(toupper(reply)=='Y');

f.close();

}

void printlist(list<list<char> > nested\_list)

{

list<list<char> >::iterator nested\_list\_itr;

// Print the nested\_list

for (nested\_list\_itr = nested\_list.begin();

nested\_list\_itr != nested\_list.end();

++nested\_list\_itr) {

cout << " [";

// normal\_list`s iterator(same type as tem

list<char>::iterator single\_list\_itr;

// pointer of each list one by one in nested list

// as loop goes on

list<char>& single\_list\_pointer = \*nested\_list\_itr;

for (single\_list\_itr = single\_list\_pointer.begin();

single\_list\_itr != single\_list\_pointer.end();

++single\_list\_itr) {

cout << " " << \*single\_list\_itr << " ";

}

cout << "]\n";

}

cout << "]";

}

void READ()

{

ifstream f("secret.dat",ios::binary | ios::in);

string1 m;

list<list<char> > nested\_list;

if(!f)

{

cout<<endl<<"FILE DOESN'T EXIST!!!!!";

return;

}

int ctr=0;

while(f.read((char\*)&m,sizeof(m)))

{

//cout<<endl<<"Record: "<<++ctr;

//m.outdata();

char l[80];

strcpy(l,m.retstr());

cout<<l<<endl;

list<char> single\_list;

for(int i=0;l[i];i++)

{

single\_list.push\_back(l[i]);

}

nested\_list.push\_back(single\_list);

}

printlist(nested\_list);

f.close();

}

int main()

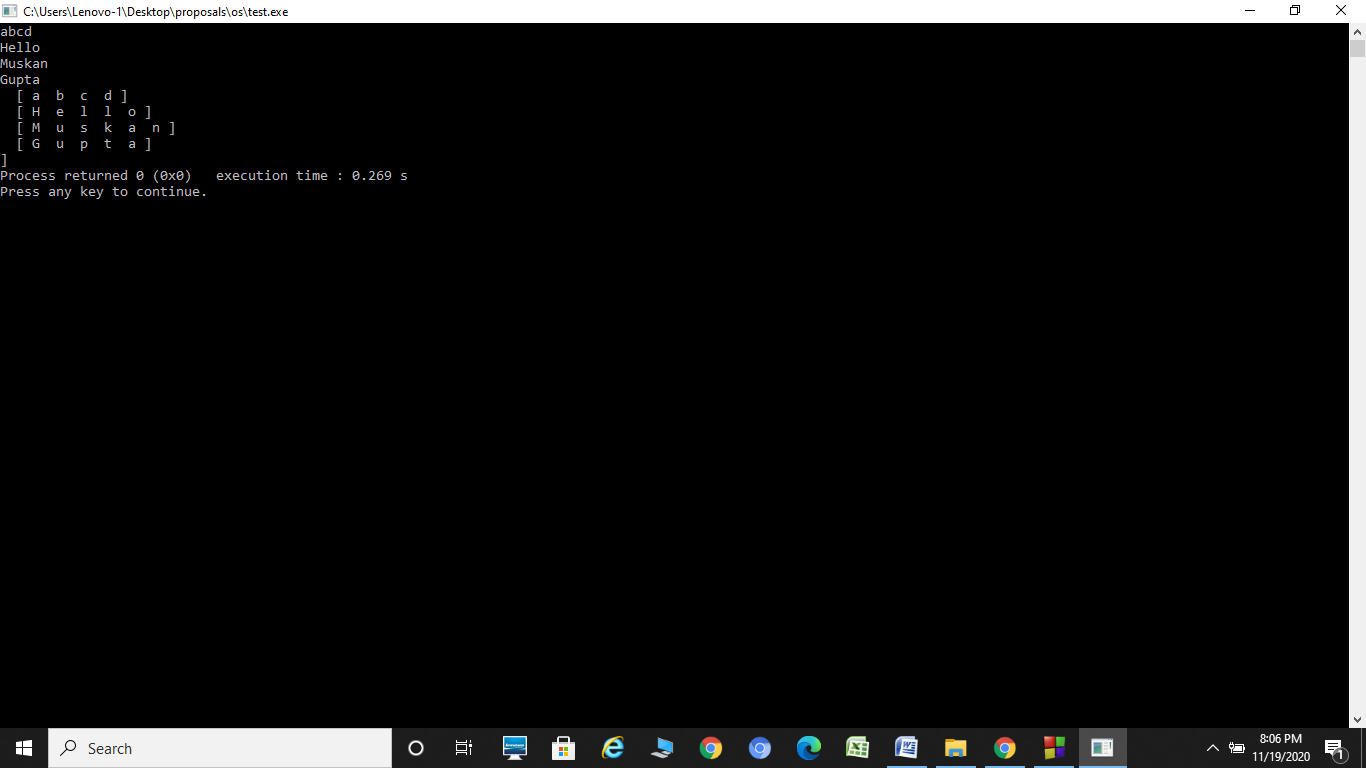
{

// WRITE();

READ();

return 0;

}



I tried this approach too

#include <bits/stdc++.h>

using namespace std;

struct node2

{

char c;

node2 \*next;

node2(char c)

{

this->c = c;

next = NULL;

}

};

struct node1

{

node2 \*str;

node1 \*next;

node1(node2 \*str)

{

this->str = str;

next = NULL;

}

};

node2 \*reverseLL2(node2 \*head2)

{

if (head2 == NULL || head2->next == NULL)

{

return head2;

}

head2->next->next = head2;

head2->next = NULL;

return head2;

}

node1 \*reverseLL1(node1 \*head1)

{

if (head1 == NULL)

{

return NULL;

}

head1->str = reverseLL2(head1->str);

if (head1->next == NULL)

{

return head1;

}

node1 \*tmp = reverseLL1(head1->next);

head1->next->next = head1;

head1->next = NULL;

return head1;

}

string getString(node2 \*head2)

{

if (head2 == NULL)

{

return "";

}

node2 \*tmp = head2;

string s = "";

while (tmp != NULL)

{

s.append(1, tmp->c);

tmp = tmp->next;

}

return s;

}

int main()

{

string text = "hello my name is mayank\nI live in Delhi\n";

ofstream fileS("secret.txt");

fileS << text;

fileS.close();

node1 \*head1 = NULL;

node1 \*tail1 = NULL;

for (int i = 0; i < text.length(); i++)

{

node2 \*head2 = NULL;

node2 \*tail2 = NULL;

if (text.at(i) != '\n')

{

node2 \*tmp = new node2(text.at(i));

if (head2 == NULL)

{

head2 = tmp;

tail2 = tmp;

}

else

{

tail2->next = tmp;

tail2 = tail2->next;

}

}

else

{

node1 \*tmp = new node1(head2);

if (head1 == NULL)

{

head1 = tmp;

tail1 = tmp;

}

else

{

tail1->next = tmp;

tail1 = tail1->next;

}

}

}

head1 = reverseLL1(head1);

ofstream fileP("public.txt");

node1 \*tmp = head1;

while (tmp != NULL)

{

string s = getString(head1->str);

fileP << s << endl;

tmp = tmp->next;

}

fileP.close();

delete head1;

delete tail1;

return 0;

}