小实验四报告

16340284 张丰露

# 实现方法

顺序线性表使用数组数据结构

链接线性表使用链表（自定义，非STL）数据结构

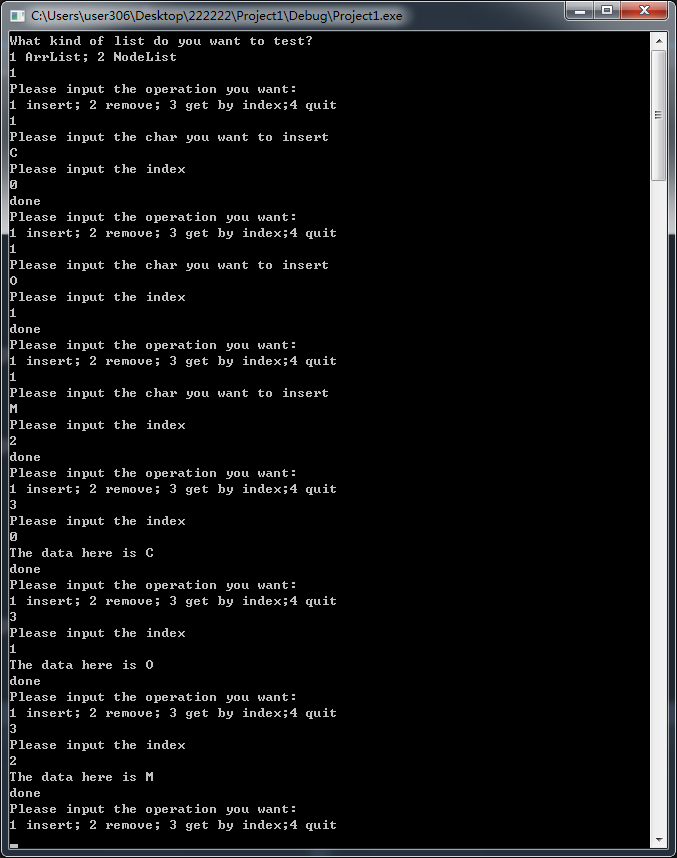
# 运行方法

请按程序提示执行。

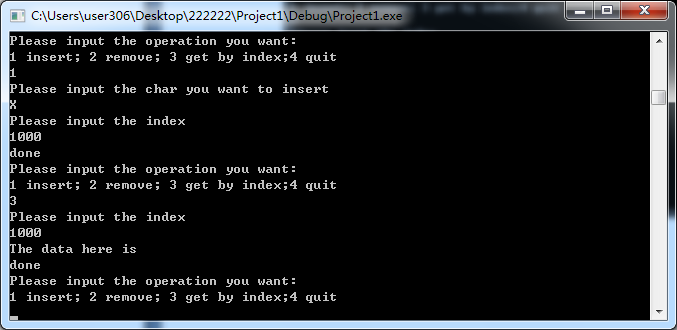
# 运行结果

### 顺序线性表

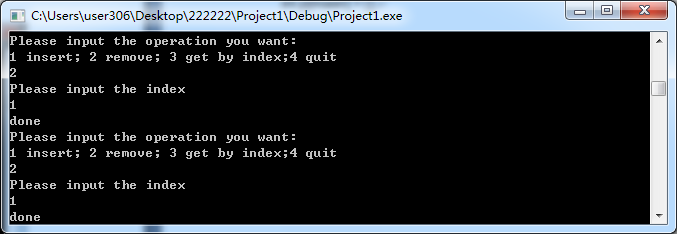
插入C,O,M并查看：

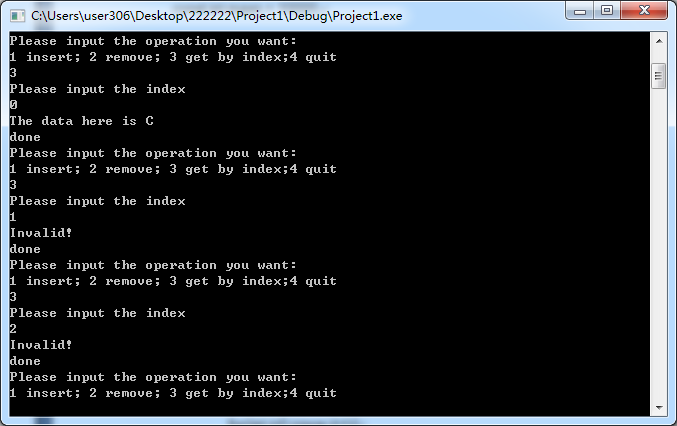


插入、移除非法值：



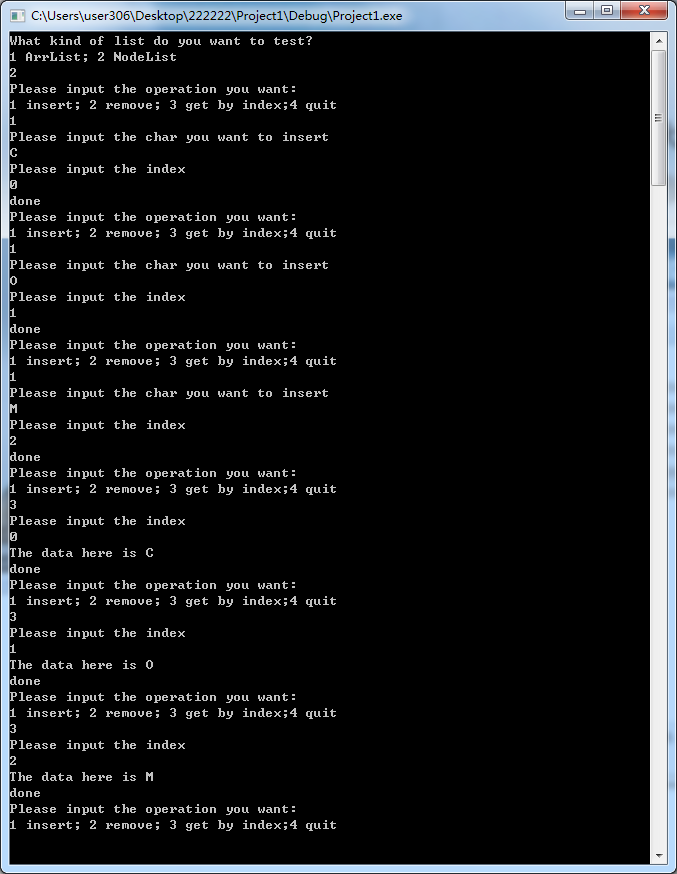
移除O,M并查看：



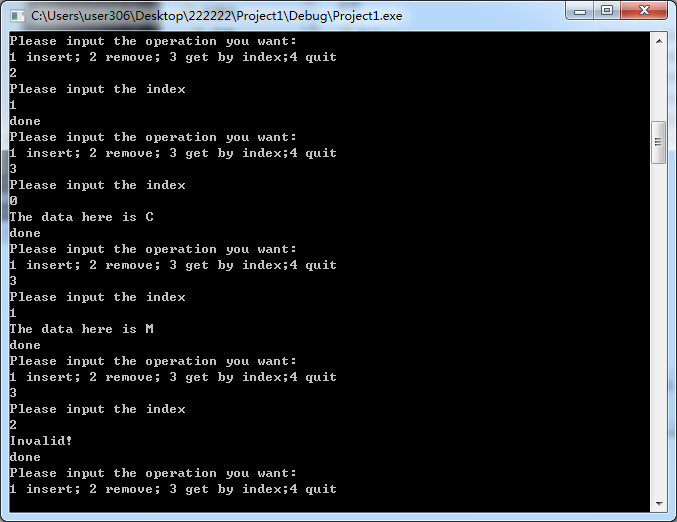


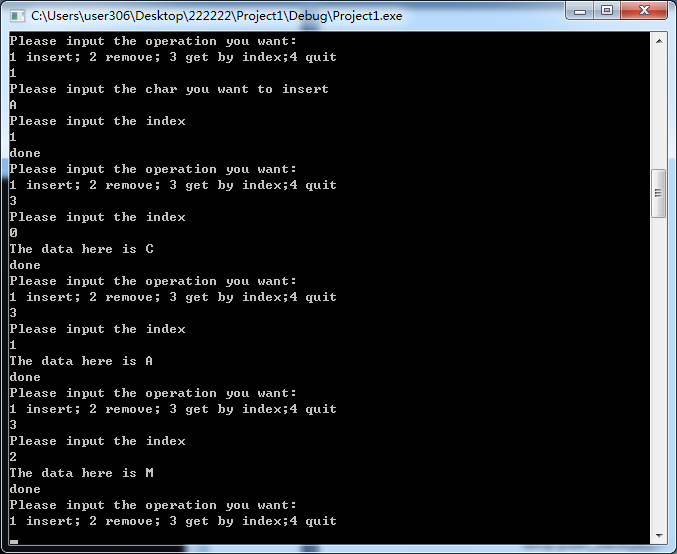
### 链接线性表

插入C,O,M并查看：

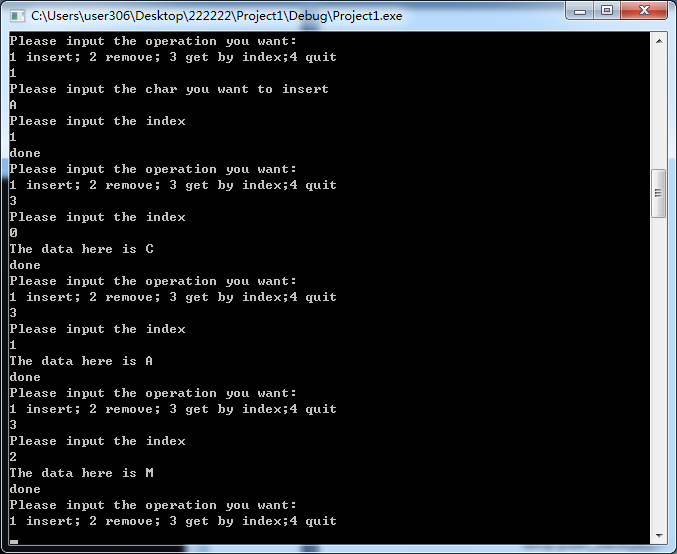


中间插入、删除：





插入，删除非法值：



# 源代码

## main.cpp

#include <iostream>

#include <cstdlib>

#include "ArrList.hpp"

#include "NodeList.hpp"

using namespace std;

void state(){

cout<<"Please input the operation you want:"<<endl;

cout<<"1 insert; 2 remove; 3 get by index;4 quit"<<endl;

}

int main(){

//说明

cout<<"What kind of list do you want to test?"<<endl;

cout<<"1 ArrList; 2 NodeList"<<endl;

int op = 0;

cin>>op;

//各项操作的选择和执行

if(op==1){

ArrList t = ArrList();

state();

cin>>op;

while(op!=0){

if(op==1){

cout<<"Please input the char you want to insert"<<endl;

char c = 0;;

cin>>c;

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

t.Insert(c,ind);

}

else if(op==2){

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

t.Remove(ind);

}

else if(op==3){

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

char ans = t.Get(ind);

if(ans == 0)

cout<<"Invalid!"<<endl;

else

cout<<"The data here is "<<t.Get(ind)<<endl;

}

cout<<"done"<<endl;

state();

cin>>op;

}

}

else if(op==2){

NodeList t = NodeList();

state();

cin>>op;

while(op!=0){

if(op==1){

cout<<"Please input the char you want to insert"<<endl;

char c = 0;;

cin>>c;

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

t.Insert(c,ind);

}

else if(op==2){

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

t.Remove(ind);

}

else if(op==3){

cout<<"Please input the index"<<endl;

int ind = 0;

cin>>ind;

char ans = t.Get(ind);

if(ans == 0)

cout<<"Invalid!"<<endl;

else

cout<<"The data here is "<<t.Get(ind)<<endl;

}

cout<<"done"<<endl;

state();

cin>>op;

}

}

system("pause");

return 0;

}

## ArrList.hpp

#include <iostream>

#include <cstring>

using namespace std;

const int maxn = 10000;

class ArrList{

public:

ArrList(){

memset(arr,0,sizeof(arr));

length = 0;

}

//插入

void Insert(char c ,int ind){

if(ind>length || ind<0)

return;

if(length >= maxn)

return;

for(int i=length-1;i>=ind;i--){

arr[i+1] = arr[i];

}

arr[ind] = c;

length++;

}

//移除

void Remove(int ind){

if(ind>=length || ind<0)

return;

if(length >= maxn)

return;

for(int i=ind;i<length;i++){

arr[i] = arr[i+1];

}

length--;

}

//得到值

char Get(int ind){

if(ind>=length || ind<0)

return 0;

return arr[ind];

}

int Length(){

return length;

}

private:

//数组数据结构

char arr[maxn];

int length;

};

## NodeList.hpp

#include <iostream>

#include <cstring>

using namespace std;

//链式数据结构

struct node{

char data;

node\* next;

node(char c, node\* n){

next = n;

data = c;

}

};

class NodeList{

public:

NodeList(){

head = NULL;

length = 0;

}

~NodeList(){

node\* pos = head;

while(pos!=NULL){

node\* temp = pos;

pos = pos->next;

delete temp;

}

}

//插入

void Insert(char c ,int ind){

if(ind>length || ind<0)

return;

if(length >= maxn)

return;

node\*\* pos = &head;

for(int i=0;i<ind;i++){

pos = &((\*pos)->next);

}

node\* temp = (\*pos);

\*pos = new node(c,temp);

length++;

}

//移除

void Remove(int ind){

if(ind>=length || ind<0)

return;

if(length >= maxn)

return;

node\*\* pos = &head;

for(int i=0;i<ind;i++){

pos = &((\*pos)->next);

}

node\* temp = (\*pos)->next;

delete(\*pos);

\*pos = temp;

length--;

}

//得到值

char Get(int ind){

if(ind>=length || ind<0)

return 0;

node\* pos = head;

for(int i=0;i<ind;i++){

pos = pos->next;

}

return pos->data;

}

int Length(){

return length;

}

private:

node \* head;

int length;

};