#include <iostream>

using namespace std;

/\*

链表的设计

1. 一个结构体代表节点

2. 一个结构体代表链表

链表里的元素

1. 头节点

2. 各种函数

链表的功能

1. 新建链表

2. 打印链表

3. 插入节点

4. 删除节点

5. 查找结点

6. 倒置链表

\*/

struct node{

int data;

node \* next;

};

struct linkedList{

int len, a;

node \* head;

node \*prev;

//一个菜单

void menu(){

int cmd;

int k = 0;

while (1){

if (k++) cout << endl;

cout << "please input the command:" << endl;

cout << "1. init a linkedList" << endl;

cout << "2. print the linkedList" << endl;

cout << "3. insert b before a" << endl;

cout << "4. delete x" << endl;

cout << "5. delete linkedList" << endl;

cout << "-1. input -1 to exit" << endl;

cin >> cmd;

if (cmd == -1) break;

switch(cmd){

case 1:

init();

break;

case 2:

print();

break;

case 3:

cout << "please input a and b" << endl;

int a, b;

cin >> a >> b;

insert (a, b);

break;

case 4:

cout << "please input x" << endl;

int x;

cin >> x;

delelteNode(x);

break;

case 5:

delelteLinkedList();

cout << "delete linkedList completed!" << endl;

break;

default:

cout << "please input the correct command" << endl;

break;

}

}

}

/\*

初始化链表

输入链表长度，然后输入各个元素的值

先初始化头指针，然后再令当前指针pre等于头指针

逐个遍历指针，prev = prev->next，最后一个结点的next为nullptr

\*/

node init(){

cout << "input the len of linkedList:" << endl;

cin >> len;

head = new node;

prev = head;

// cout << "please input the data of every node:" << endl;

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

// cin >> a;

prev->data = i + 3;

prev->next = new node;

prev = prev->next;

}

prev->next = nullptr;

cout << "A new linkedList is inited!" << endl;

}

/\*

打印链表

从头指针开始遍历，当前指针prev等于头指针head,然后每次输出当前指针的data值

接着prev = prev->next

直到prev->next为空指针时终止

\*/

void print(){

prev = head;

if (head == nullptr){

cout << "the linkedList do not exist!" << endl;

return;

}

while (prev->next != nullptr){

cout << prev->data << " ";

prev = prev->next;

}

cout << endl;

}

/\*

查找某个值为x的结点的上一个结点

\*/

node \*find(int x){

prev = head;

if (head->data == x){

return (node\*)-1;

}

else{

while (prev->next->data != x && prev != nullptr){

prev = prev->next;

}

return prev;

}

}

/\*

在值为a的元素前面插入b

\*/

void insert(int a, int b){

prev = find (a);

if (prev == nullptr){

cout << "The linkedList do not exist " << a << " " << endl;

return;

}

node \*q = new node;

if (prev == (node\*)-1){

q->data = b;

q->next = head;

head = q;

}

else{

q->data = b;

q->next = prev->next;

prev->next = q;

}

}

/\*

删除某个结点

\*/

void delelteNode(int x){

prev = find(x);

if (prev == nullptr){

cout << "The linkedList do not exist " << x << " " << endl;

return;

}

if (prev == (node\*)-1){

prev = head;

head = head->next;

delete prev;

}

else{

node\* t = prev->next;

prev->next = t->next;

delete t;

}

}

/\*

删除整个链表

\*/

void delelteLinkedList(){

prev = head;

while (prev != nullptr){

node \*t = prev;

prev = prev->next;

delete t;

}

head = nullptr;

}

void reverse(){

}

};

int main(){

linkedList a;

a.menu();

return 0;

}