#include <iostream>

using namespace std;

template<class any>

class list

{

private:

template<class any>

struct node

{

any data;

node\* pNext;

node(any data = any(), node\* pNext = nullptr)

{

this->data = data;

this->pNext = pNext;

}

};

int size;

node<any>\* head;

public:

list();

~list();

any& operator[](const int idx);

int getSize() { return size; }

void pushBack(any data);

void removeByIdx(int idx);

void popBack();

void insert(any val, int idx);

};

template<class any>

list<any>::list()

{

size = 0;

head = nullptr;

}

template<class any>

list<any>::~list()

{

cout << "\ndtor\n";

}

template<class any>

any& list<any>::operator[](const int idx)

{

int counter = 0;

node<any>\* currentElement = this->head;

while (currentElement != nullptr)

{

if (counter == idx)

{

return currentElement->data;

}

currentElement = currentElement->pNext;

counter++;

}

}

template<class any>

void list<any>::pushBack(any data)

{

if (head == nullptr)

{

head = new node<any>(data);

}

else

{

node<any>\* currentElement = this->head;

while (currentElement->pNext != nullptr)

{

currentElement = currentElement->pNext;

}

currentElement->pNext = new node<any>(data);

}

size++;

}

template<class any>

void list<any>::removeByIdx(int idx)

{

if (idx == 0)

{

node<any>\* temp = head;

head = head->pNext;

delete temp;

size--;

}

else

{

node<any>\* previousElement = this->head;

for (int i = 0; i < idx - 1; i++)

previousElement = previousElement->pNext;

node<any>\* toDelete = previousElement->pNext;

previousElement->pNext = toDelete->pNext;

delete toDelete;

size--;

}

}

template<class any>

void list<any>::popBack()

{

removeByIdx(size - 1);

}

template<class any>

void list<any>::insert(any val, int idx)

{

if (idx == 0)

{

head = new node<any>(val, head);

size++;

}

else

{

node<any>\* previousElement = this->head;

for (int i = 0; i < idx-1; i++)

previousElement = previousElement->pNext;

node<any>\* newNode = new node<any>(val, previousElement->pNext);

previousElement->pNext = newNode;

size++;

}

}

int main()

{

list<int> lst;

for (int i = 0; i < 5; i++)

lst.pushBack(10);

cout << endl;

cout << "size = " << lst.getSize();

cout << endl;

for (int i = 0; i < lst.getSize(); i++)

cout << i <<"element: "<< lst[i] << "\n";

cout << "\nlst.insert(44, 2);\n";

lst.insert(44, 2);

cout << endl;

for (int i = 0; i < lst.getSize(); i++)

cout << i << "element: " << lst[i] << "\n";

cout << "\nlst.pushBack(77);\n";

lst.pushBack(77);

cout << endl;

for (int i = 0; i < lst.getSize(); i++)

cout << i << "element: " << lst[i] << "\n";

cout << "\nlst.removeByIdx(5);\n";

lst.removeByIdx(5);

cout << endl;

for (int i = 0; i < lst.getSize(); i++)

cout << i << "element: " << lst[i] << "\n";

cout << "\nlst.popBack();\n";

lst.popBack();

cout << endl;

for (int i = 0; i < lst.getSize(); i++)

cout << i << "element: " << lst[i] << "\n";

}