#include <iostream>

#include <vector>

#include <ctime>

using namespace std;

struct Item // Динамический список

{

int data; // Элемент

Item\* prev = NULL; // Указатель на предыдущий

};

class Stack

{

private:

Item\* Head; // Указатель на первый элемент

Item\* Tail; // Указатель на последний элемент

int size; // Кол-во элементов

public:

Stack(); // Конструктор

void add(int a); // Добавление элемента (a)

int value(); // Значение последнего элемента

void del(); // Удалить элемент

int get(int x); // Взять (x)ый элемент

void sort(); // Сортировка элементов

void swap(int a, int b); // Поменять элементы (a) и (b) местами

void set(int n, int m); // Заменить (n)ый элемент на значение m

};

Stack::Stack()

{

size = 0;

Item\* Head = NULL;

Item\* Tail = NULL;

}

void Stack::add(int m)

{

Item\* temp = new Item;

temp->data = m;

temp->prev = Tail;

if (size == 0)

Head = Tail = temp;

else

Tail = temp;

size++;

}

int Stack::value()

{

return Tail->data;

}

void Stack::del()

{

Item\* Del = Tail;

Tail = Tail->prev;

delete Del;

size--;

}

int Stack::get(int n)

{

Stack temp;

while (size - 1 != n)

{

temp.add(value());

del();

}

int a = value();

while (temp.size != 0)

{

add(temp.value());

temp.del();

}

return a;

}

void Stack::sort()

{

for (int j = 0; j < size; j++)

{

for (int i = size / 2 - 1 - j / 2; i > -1; i--)

{

if (2 \* i + 2 <= size - 1 - j)

{

if (get(2 \* i + 1) > get(2 \* i + 2))

{

if (get(i) < get(2 \* i + 1))

{

swap(i, 2 \* i + 1);

}

}

else

if (get(i) < get(2 \* i + 2))

{

swap(i, 2 \* i + 2);

}

}

else

if (2 \* i + 1 <= size - 1 - j)

if (get(i) < get(2 \* i + 1))

swap(i, 2 \* i + 1);

}

swap(0, size - 1 - j);

}

}

void Stack::swap(int a, int b)

{

int x = get(a);

int y = get(b);

set(b, x);

set(a, y);

}

void Stack::set(int n, int m)

{

Stack temp;

while (size - 1 != n)

{

temp.add(value());

del();

}

del();

add(m);

while (temp.size != 0)

{

add(temp.value());

temp.del();

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(0));

int N = 10;

Stack mystack;

for (int i = N - 1; i >= 0; i--)

mystack.add((int)rand() % 200 - 100);

cout << "Первоначальный список:\n";

for (int i = N - 1; i > -1; i--)

cout << " " << mystack.get(i);

cout << endl;

mystack.sort();

cout << "Отсортированный список:\n";

for (int i = N - 1; i > -1; i--)

cout << " " << mystack.get(i);

cout << endl;

return 0;

}