#include <iostream>

#include <iomanip>

using namespace std;

struct Node {

int data;

Node\* prev;

};

struct Stack {

Node\* top;

int amount;

};

Stack\* createStack();

void deleteStack(Stack\*\*);

void printStack(Stack\*);

void push(Stack\*, int);

int pop(Stack\*);

void sort(Stack\*);

Stack\* createStack() {

Stack\* st = new Stack;

st->top = nullptr;

st->amount = 0;

return st;

}

void deleteStack(Stack\*\* st) {

Node\* last = (\*st)->top;

Node\* second = nullptr;

while (last != nullptr) {

second = last;

last = last->prev;

delete second;

(\*st)->amount--;

}

delete\* st;

\*st = nullptr;

}

void printStack(Stack\* st) {

if (st == nullptr)

{

cout << endl << "Стек пустой";

return;

}

Node\* n = st->top;

while (n != nullptr) {

cout << n->data << setw(15);

n = n->prev;

}

}

void push(Stack\* st, int newNumb) {

Node\* n = new Node;

n->prev = st->top;

n->data = newNumb;

st->top = n;

st->amount++;

}

int pop(Stack\* st) {

if (st->top == nullptr) {

cout << "В стеке отсутствуют элементы";

return 1;

}

else {

Node\* n = st->top;

st->top = n->prev;

int numb = n->data;

delete n;

st->amount--;

return numb;

}

}

void sort(Stack\* st) {

int max = st->top->data;

Node\* temp = st->top;

Node\* prevToDelete = nullptr;

Node\* toDelete = st->top;

while (temp->prev != nullptr) {

if (temp->prev->data > max) {

max = temp->prev->data;

prevToDelete = temp;

toDelete = temp->prev;

}

temp = temp->prev;

}

if (max == st->top->data) {

st->top = st->top->prev;

delete toDelete;

return;

}

prevToDelete->prev = toDelete->prev;

delete toDelete;

return;

}

int main() {

setlocale(LC\_ALL, "Russian");

Stack\* st = createStack();

int k, numb, amount, temp = 0;

cout << "Сколько чисел вы хотите ввести: ";

cin >> amount;

while (temp != amount) {

cout << "Введите ваше число: ";

cin >> numb;

push(st, numb);

temp++;

}

cout << "Ваш стек: ";

printStack(st);

cout << endl;

sort(st);

cout << "Ваш отсортированный стек: ";

printStack(st);

deleteStack(&st);

return 0;

}