Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №15**

По дисциплине «Основы алгоритмизации и программирования»

Тема: «Программирование с использованием однонаправленных списков типа “Стек” и “Очередь”»

**Выполнил:**

Студент 1 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Гирель Т.Н

Брест 2022

**Цель работы:** Приобретение навыков работы в использовании однонаправленных списков типа “Стек” и “Очередь” на С/C++.

**Вариант №1**

**Текст задания №1**

**Создать стек с числами в диапазоне от –50 до +50. После создания стека преобразовать стек в два стека. Первый должен содержать только положительные числа, а второй – отрицательные. В конце работы все стеки должны быть удалены.**

**Текст программы**

#include <iostream>

#include <malloc.h>

using namespace std;

struct stack { short inf; stack\* s; } \*sp,\*pls,\*mins;

stack\* addstack(stack\* st, int inf) {

stack\* new\_stack = new stack;

new\_stack->inf = inf;

new\_stack->s = st;

return new\_stack;}

void outstack(stack\* st) {

cout << endl;

while (st != NULL) {

cout << st->inf << " ";

st = st->s;}

cout << endl;}

void razdelenie(stack\* st, stack\* &plus, stack\* &minus) {

stack\* buff\_plus=0, \* buff\_minus=0;

while (st != 0) {

if (st->inf > 0) {

buff\_plus = addstack(buff\_plus, st->inf);

st = st->s;

continue;}

if (st->inf < 0) {

buff\_minus = addstack(buff\_minus, st->inf);

st = st->s;

continue;}

st = st->s;}

while (buff\_plus != 0) {

plus = addstack(plus, buff\_plus->inf);

buff\_plus=buff\_plus->s;}

while (buff\_minus != 0) {

minus = addstack(minus, buff\_minus->inf);

buff\_minus=buff\_minus->s;}}

int main(){

int a;

cout << "Введите весь стек(числа от -50 до +50): ";

do {

cin >> a;

if (a > 50 || a < -50) continue;

else sp = addstack(sp, a);

} while (cin.peek()!='\n');

cout << "Начальный стек: ";

outstack(sp);

razdelenie(sp, pls, mins);

cout << "Положительный стек: ";

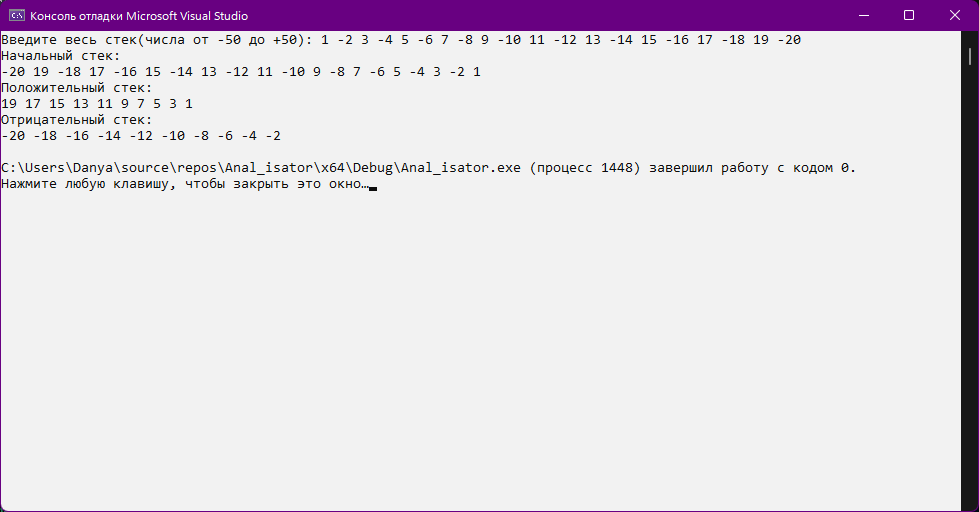
outstack(pls);

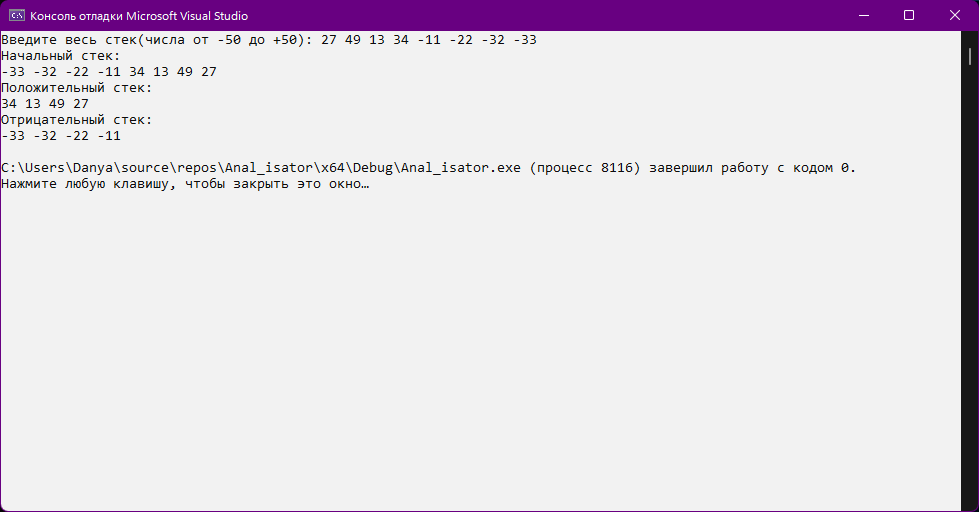
cout << "Отрицательный стек: ";

outstack(mins);

return 0;}

**Результат работы**





**Текст задания №2**

**Создать однонаправленную очередь с числами в диапазоне от –50 до +50.**

**После создания очереди удалить из очереди все четные числа. В конце работы все**

**очереди должны быть удалены.**

**Текст программы**

#include <iostream>

using namespace std;

struct ochered {

int info;

ochered\* next;

}\*och;

void addoch(ochered\* &och, int info) {

if (och == NULL) {

och = new ochered;

och->info = info;

och->next = NULL;

return;}

ochered\* oc = och;

while (oc->next != NULL) oc = oc->next;

ochered\* nextoc = new ochered;

nextoc->info = info;

nextoc->next = NULL;

oc->next = nextoc;

return;}

void outoch(ochered\* och) {

cout << endl;

while (och != NULL) {

cout << och->info << " ";

och = och->next;}

cout << endl;};

void individ(ochered\*& och) {

ochered\* oc = och,\*buff\_oc=0,\*prevbuff=0;

if (och->info % 2 == 0) {

och = och->next;

delete oc;

oc = och;}

while (oc != NULL) {

if (oc->next != NULL && (oc->next)->info % 2 == 0) {

buff\_oc = oc->next;

while (buff\_oc!=NULL&&buff\_oc->info % 2 == 0) {

prevbuff = buff\_oc;

buff\_oc = buff\_oc->next;

delete prevbuff;}

oc->next = buff\_oc;

oc = oc->next;

continue;}

else { oc = oc->next; }}}

int main() {

int a;

cout << "Введите всю очередь(числа от -50 до +50): ";

do {

cin >> a;

if (a > 50 || a < -50) continue;

else addoch(och, a);

} while (cin.peek() != '\n');

cout << "Очередь до: ";

outoch(och);

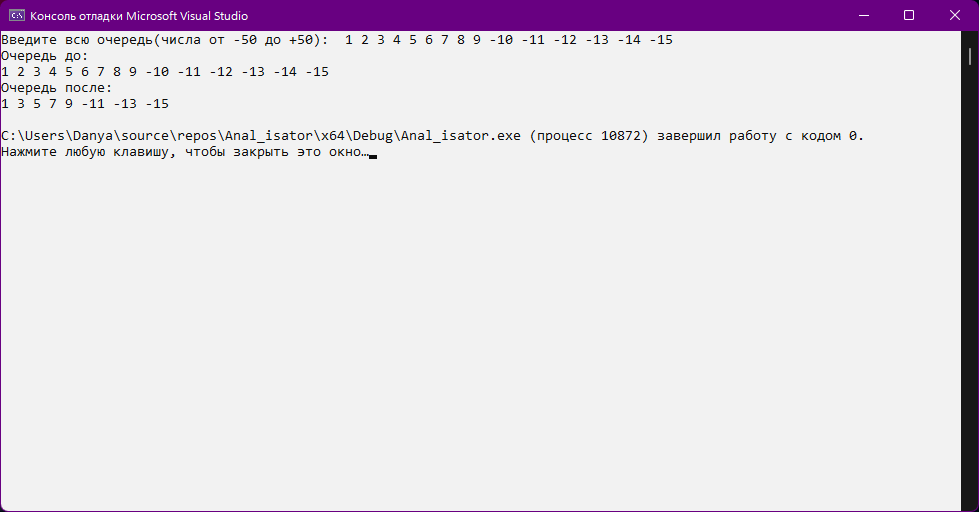
individ(och);

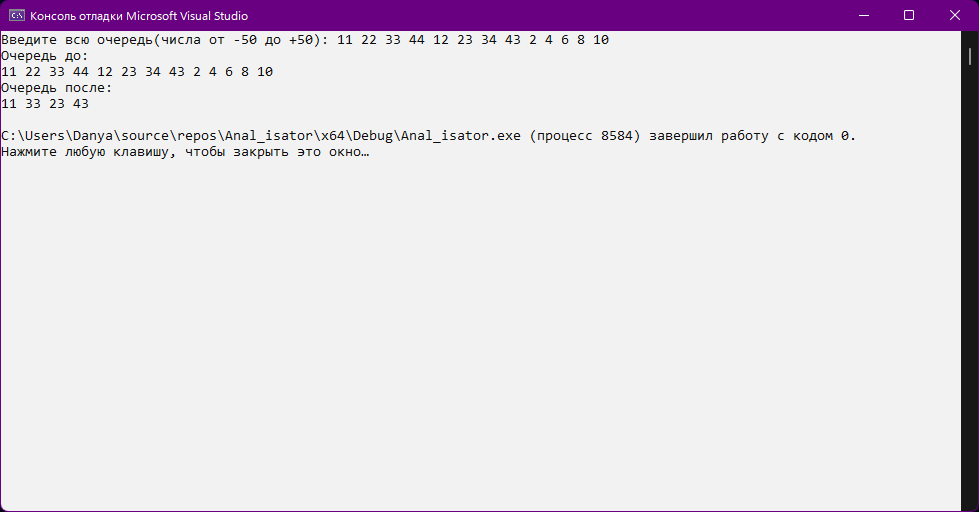
cout << "Очередь после: ";

outoch(och);

return 0;}

**Результат работы**





**Вывод:** Приобрёл навыки работы в использовании однонаправленных списков типа “Стек” и “Очередь” на С/C++.