Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра микро- и наноэлектроники

**Лабораторная работа №4**

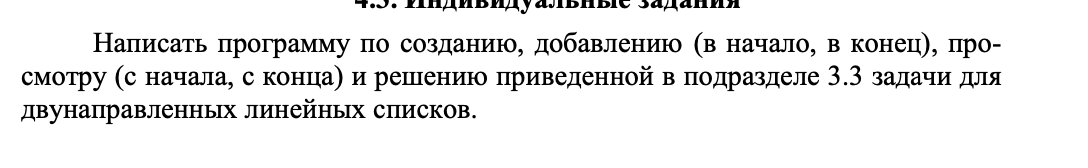
**«2-связанные списки»**

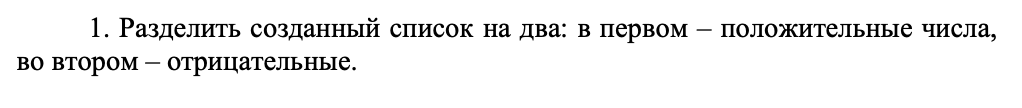
**Вариант 1**

|  |  |
| --- | --- |
| Выполнил: | студент группы 142701  Ахремчик Никита Андреевич |
| Проверил: |  |

Минск 2022

**Задачи:** изучить возможности работы со списками, организован- ными в виде очереди.

**Индивидуальное задание**



Текст программы:

#**include** <iostream>

#**include** <stdio.h>

#**include** <string>

#**include** <time.h>

**const** **char** EN [] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

**const** **char** en [] = "abcdefghijklmnopqrstuvwxyz";

**const** **char** num [] = "1234567890"; // добавить символов

**struct** queue{

queue \*prev, \*next;

**int** data;

} \*begin, \*end, \*begin2, \*end2, \*t, \*t1, \*b1, \*b2, \*e1, \*e2;

**typedef** **void** (\*tdinsert)(queue \*\*, **int**);

**typedef** **void** (\*tdinsertend)(queue \*\*, **int**);

**typedef** **void** (\*tdcreate)(queue \*\*, queue \*\*, **int**);

**void** **create**(queue \*\*b, queue \*\*e, **int** data){

t = **new** queue;

t->data = data;

t->next = t->prev = nullptr;

\*b = \*e = t;

}

**void** **insert\_begin**(queue \*\*b, **int** data){

t = **new** queue;

t->data = data;

t->prev = nullptr;

t->next = \*b;

(\*b)->prev = t;

\*b = t;

}

**void** **insert\_end**(queue \*\*e, **int** data){

t1 = **new** queue;

t1->data = data;

t1->next = nullptr;

t1->prev = \*e;

(\*e)->next = t1;

\*e = t1;

}

**void** **output\_begin**(queue \*t){

printf("Элементы очереди СНАЧАЛА: ");

**while**(t != nullptr){

printf("%i ", t->data);

t = t->next;

}

puts("");

}

**void** **output\_end**(queue \*t1){

printf("Элементы очереди С КОНЦА: ");

**while**(t1 != nullptr){

printf("%i ", t1->data);

t1 = t1->prev;

}

puts("");

}

**void** **del**(queue \*\*q) {

**while**(\*q != nullptr){

t = \*q;

\*q = (\*q)->next;

**delete** t;

}

}

**void** **chooses**(queue \*\*b, queue \*\*e, queue \*\*b1, queue \*\*e1, queue \*\*b2, queue \*\*e2, tdinsert insert\_begin, tdinsertend insert\_end, tdcreate create, std::string cond="b"){

**do**{

**if** (cond == "b"){

**if** ((\*b)->data >= 0){

**if** (\*b1 == nullptr && \*e1 == nullptr) create(b1, e1, (\*b)->data);

**else** insert\_begin(b1, (\*b)->data);

}

**else**{

**if** (\*b2 == nullptr && \*e2 == nullptr) create(b2, e2, (\*b)->data);

**else** insert\_begin(b2, (\*b)->data);

}

\*b = (\*b)->next;

}

**else**{

**if** ((\*e)->data >= 0){

**if** (\*b1 == nullptr && \*e1 == nullptr) create(b1, e1, (\*e)->data);

**else** insert\_end(e1, (\*e)->data);

}

**else**{

**if** (\*b2 == nullptr && \*e2 == nullptr) create(b2, e2, (\*e)->data);

**else** insert\_end(e2, (\*e)->data);

}

\*e = (\*e)->prev;

}

}**while** (\*b != nullptr && \*e != nullptr);

}

**int** **main**() {

srand(time(nullptr));

**int** num, data;

**bool** success = true;

printf("Длина стека: ");

**while**(success){

std::cin >> num;

**if** (!std::cin.good()){

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

printf("Вы ввели не число! Повторите ввод: ");

}

**else**

success = false;

puts("");

}

**for** (**int** i=0; i<num; i++){

data = rand() % 100 - 50;

**if** (begin == nullptr && end == nullptr) create(&begin, &end, data);

**else** insert\_begin(&begin, data);

}

output\_begin(begin);

**for** (**int** i=0; i<num; i++){

data = rand() % 100 - 50;

**if** (begin2 == nullptr && end2 == nullptr) create(&begin2, &end2, data);

**else** insert\_end(&end2, data);

}

output\_end(end);

puts("");

std::string cond;

printf("Введите условие для разделения очереди(b/e): ");

std::cin >> cond;

**if** (cond == "b") chooses(&begin, &end, &b1, &e1, &b2, &e2, &insert\_begin, &insert\_end, &create, cond);

**if** (cond == "e") chooses(&begin2, &end2, &b1, &e1, &b2, &e2, &insert\_begin, &insert\_end, &create, cond);

**if** (cond == "b"){

output\_begin(b1);

output\_begin(b2);

}

**if** (cond == "e"){

output\_end(e1);

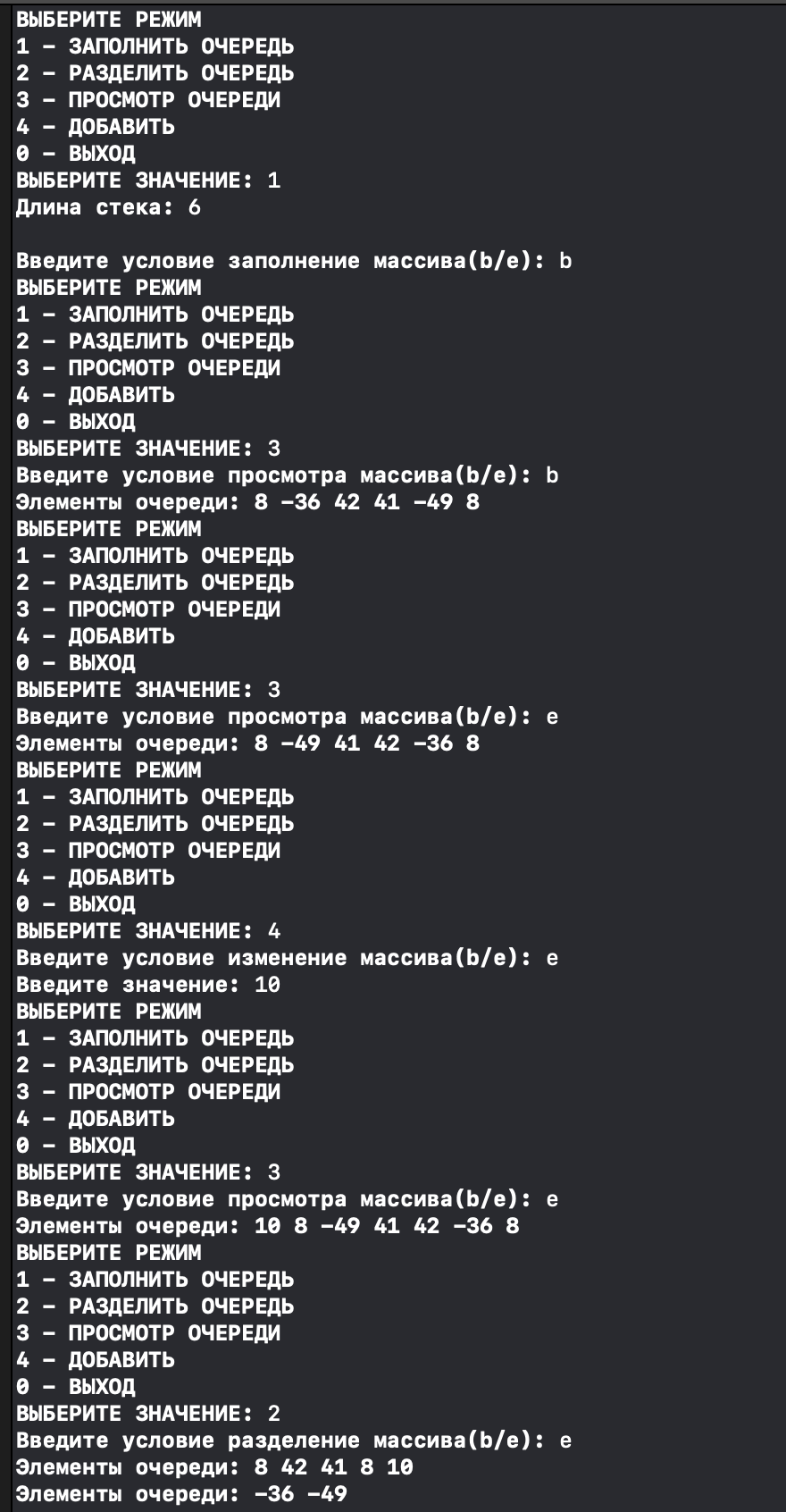
output\_end(e2);

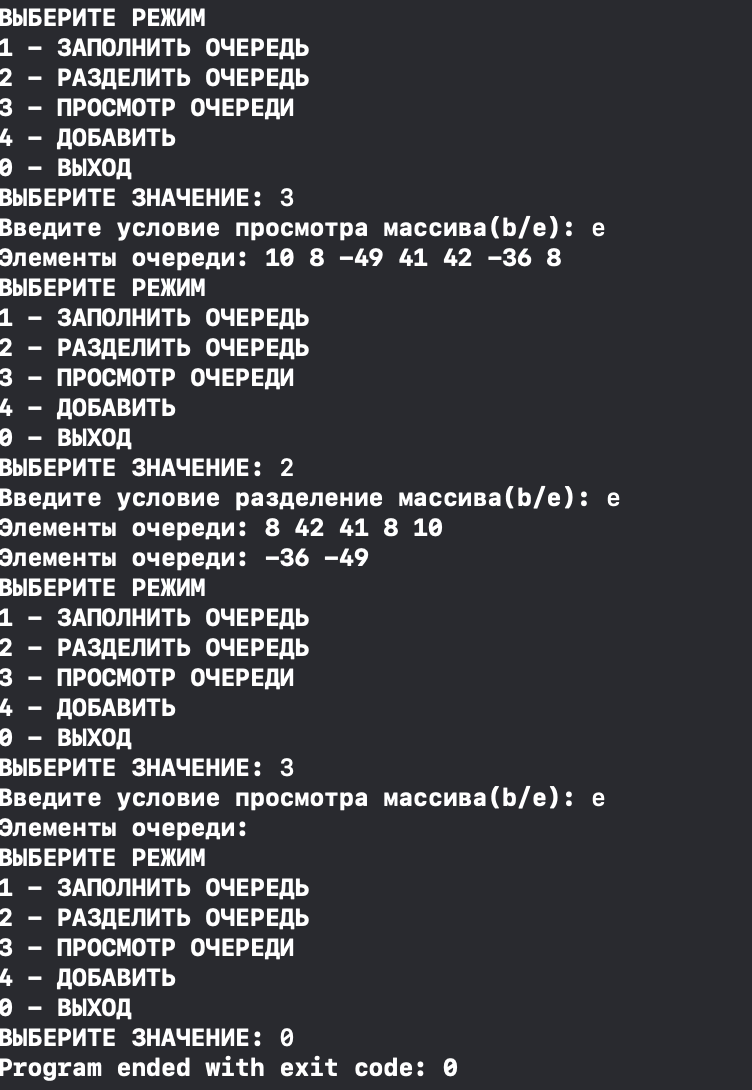
}

**return** 0;

}

**Результат вывода программы:**





**Выводы:** Проведя данную лабораторную работу, я научился работать с использованием 2-связанных списков в С++. Также закрепил знания по прошлым темам.