**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №12**

**«**По дисциплине «Основы алгоритмизации и программирования»

**Тема:**

Списки.

Выполнила:

Студентка группы ИВТ-19-1б

Еремеева Полина Алексеевна

Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2020

**Постановка задачи**

Однонаправленный список:Тип информационного поля int.

Удалить из списка первый элемент с четным информационным полем.

Двунаправленный список: Тип информационного поля char\*.

Добавить в список элемент с заданным номером.

**Код**

#include <iostream>

**using** **namespace** std;

**struct** Node

{

**int** data; //Данные

Node\* next;

};

**struct** Node2

{ //Узел

**char**\* data;

Node2\* next;

Node2\* prev;

};

Node\* make\_list(**int** n);

Node2\* make\_list2(**int** n);

**int** main()

{

setlocale(LC\_ALL, "ru");

cout << "Одноаправленый список.\n";

**int** n;

cout << "Введите длину списка: ";

cin >> n;

Node\* list = make\_list(n);

**bool** dltd = **false**;

**if** (list->data % 2 == 0)

{

Node\* chetn = list;

list = list->next;

**delete** chetn;

dltd = **true**;

n--;

}

Node\* nazvanie = list;

**for** (**int** i = 2; i <= n && !dltd; i++)

{

**if** (nazvanie->next->data % 2 == 0)

{

Node\* chetn = nazvanie->next;

nazvanie->next = nazvanie->next->next;

**delete** chetn;

dltd = **true**;

n--;

}

nazvanie = nazvanie->next;

}

nazvanie = list;

**for** (**int** i = 1; i <= n; i++)

{

cout << nazvanie->data << ' ';

nazvanie = nazvanie->next;

}

cout << "\n\nДвунаправленый список.\n";

cout << "Введите длину списка: ";

cin >> n;

Node2\* list2 = make\_list2(n);

Node2\* nazvanie2 = list2;

**int** number;

**char** al;

**do**

{

cout << "Введите номер элемента: \n";

cin >> number;

} **while** (number > n);

cout << "Введите элемент: \n";

cin >> al;

**for** (**int** i = 1; i < number; i++)

{

nazvanie2 = nazvanie2->next;

}

Node2\* zap = nazvanie2;

Node2\* new\_al = **new** Node2;

new\_al->data = &al;

new\_al->next = zap;

new\_al->prev = zap->prev;

zap->prev = new\_al;

zap->prev->prev->next = new\_al;

n++;

**for** (**int** i = 1; i <= n; i++)

{

cout << \*(list2->data) << ' ';

**if** (list2->next != **NULL**)

{

list2 = list2->next;

**delete** list2->prev;

}

**else**

{

**delete** list2;

}

}

}

Node\* make\_list(**int** n)

{ //Создание списка

Node\* beg = **NULL**, \* p, \* r;

p = **new** Node;

cin >> p->data;

beg = p;

**for** (**int** i = 2; i <= n; i++)

{

r = **new** Node;

cin >> r->data;

p->next = r;

r->next = **NULL**;

p = r;

}

**return** beg;

}

Node2\* make\_list2(**int** n)

{ //Создание списка

Node2\* beg = **NULL**, \* p, \* r;

p = **new** Node2;

p->data = **new** **char**;

cin >> \*(p->data);

beg = p;

p->prev = **NULL**;

**for** (**int** i = 2; i <= n; i++)

{

r = **new** Node2;

r->data = **new** **char**;

cin >> \*(r->data);

p->next = r;

r->next = **NULL**;

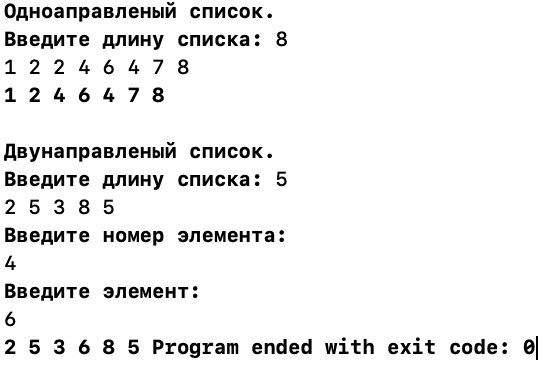
r->prev = p;

p = r;

}

**return** beg;

}

**Тестирование**