Федеральное государственное автономное

образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ЛАБОРАТОРНАЯ РАБОТА №6**

# Тема: “Разработка программ обработки двусвязных линейных списков”

# Вариант 2

Преподаватель

Студент КИ

Красноярск 2013

**Цель работы:**

приобретение навыков обработки односвязных линейных списков.

**Задача:**

Удалить из списка узел, содержащий заданное натуральное число x. Оформить как функцию поиск в списке узла, содержащего заданное натуральное число x.

**Порядок выполнения работы:**  
  
В соответствии с поставленной задачей необходимо разработать и отладить программу обработки двусвязных линейных списков, подготовить отчет и защитить лабораторную работу преподавателю. Во многих вариантах данного задания ставится задача либо перестановки каких-либо узлов списка, либо разбиения или объединения списков. В этих задачах ни в коем случае не допускается динамическое создание новых узлов и простое копирование в них информации из уже существующих узлов списков. Все операции по перемещению узлов внутри списка или между списками обязательно должны производиться с помощью переназначения указателей на узлы.

#include <stdio.h>

#include <conio.h>

struct node

{

int g;

node \*rlink, \*llink;

};

void llist(node \*left)

{

node \*k = left;

while (k != NULL)

{

printf("%d\n", k->g);

k = k->rlink;

}

}

void rlist (node \*rigth)

{

node \*k = rigth;

while (k != NULL)

{

printf("%d\n", k->g);

k = k->llink;

}

}

void main ()

{

node \*k,\*q,\*left,\*rigth;

int w;

printf("vvedite chislo");

scanf("%d",&w);

k = new node;

k->g = w;

k->rlink = NULL;

k->llink = NULL;

rigth = k;

q = k;

printf("vvedite chisla");

scanf("%d",&w);

while (!feof(stdin))

{

k = new node;

k->g = w;

k->rlink = q;

q->llink = k;

q = k;

printf("vvedite chislo");

scanf("%d", &w);

}

q->llink = NULL;

left = q;

puts("sleva napravo");

llist(left);

puts("sprava nalevo");

rlist(rigth);

node \*i;

int x;

printf("vvedite x\n");

scanf("%d", &x);

i = k = left;

while (k!=NULL)

{

if(k->g == x)

{

if(k == left)

{

q = k;

left = k->rlink;

left->llink = NULL;

i = k = left;

delete q;

}

else if (k == rigth)

{

q = k;

rigth = k->llink;

rigth->rlink = NULL;

k = NULL;

delete q;

}

else

{

q = k;

i->rlink = k->rlink;

k->rlink->llink = i;

delete q;

k = i->rlink;

}

}

else

{

i = k;

k = k->rlink;

}

}

puts("conclusion after removing x ");

puts("sleva napravo");

llist(left);

puts("sprava nalevo");

rlist(rigth);

getch();

}