Министерство цифрового развития, связи и массовых коммуникаций РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики» (СибГУТИ)

Кафедра телекоммуникационных сетей и вычислительных средств

**Отчет**

на тему

## **"**№16. Многосвязные списки. **Вариант 15"**

Выполнил: студент I курса

ИВТ, гр. ИА-132

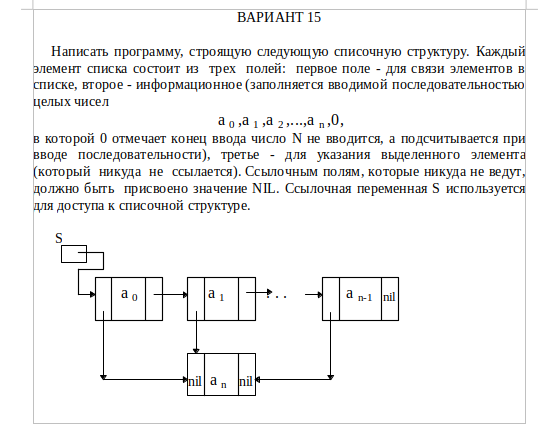
Кулаков К.В.

Проверил:

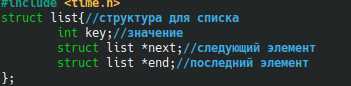
Вейлер А. И

Новосибирск 2022

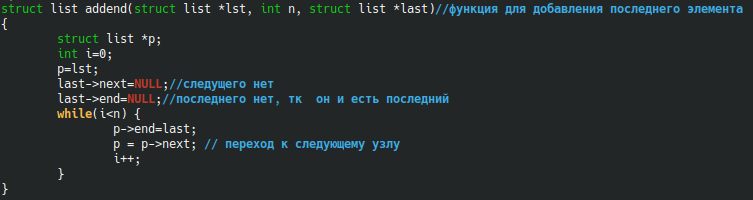
Вариант 15



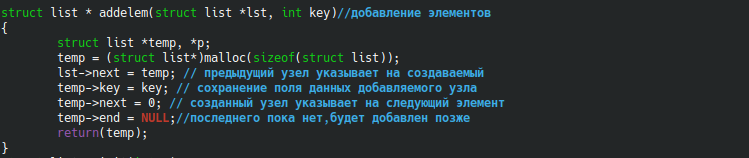
Структура:



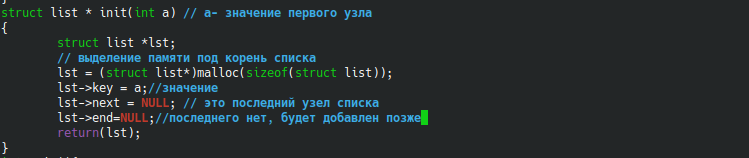
Функция для вставки последнего элемента:



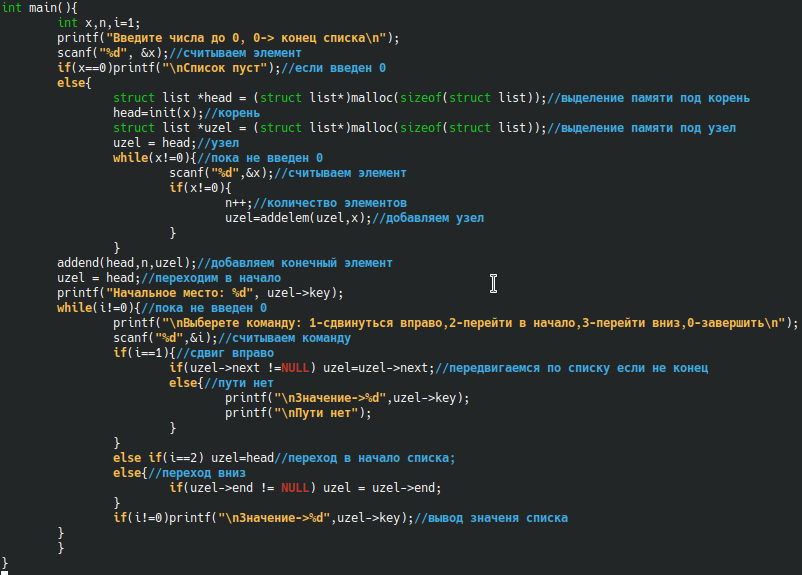
Функция для добавления нового элемента:



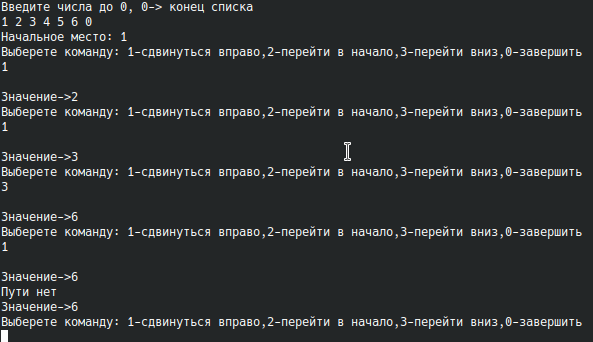
Функция для инициализации корня списка:



Главная функция:



Результат работы:



------------------------------------------------------------------------------------

**Файл main.c**

------------------------------------------------------------------------------------

#include <stdio.h>//библиотека ввода вывода

#include <stdlib.h>

#include <time.h>

struct list{//структура для списка

int key;//значение

struct list \*next;//следующий элемент

struct list \*end;//последний элемент

};

struct list addend(struct list \*lst, int n, struct list \*last)//функция для добавления последнего элемента

{

struct list \*p;

int i=0;

p=lst;

last->next=NULL;//следущего нет

last->end=NULL;//последнего нет, тк он и есть последний

while(i<n) {

p->end=last;

p = p->next; // переход к следующему узлу

i++;

}

}

struct list \* addelem(struct list \*lst, int key)//добавление элементов

{

struct list \*temp, \*p;

temp = (struct list\*)malloc(sizeof(struct list));

lst->next = temp; // предыдущий узел указывает на создаваемый

temp->key = key; // сохранение поля данных добавляемого узла

temp->next = 0; // созданный узел указывает на следующий элемент

temp->end = NULL;//последнего пока нет,будет добавлен позже

return(temp);

}

struct list \* init(int a) // а- значение первого узла

{

struct list \*lst;

// выделение памяти под корень списка

lst = (struct list\*)malloc(sizeof(struct list));

lst->key = a;//значение

lst->next = NULL; // это последний узел списка

lst->end=NULL;//последнего нет, будет добавлен позже

return(lst);

}

int main(){

int x,n,i=1;

printf("Введите числа до 0, 0-> конец списка\n");

scanf("%d", &x);//считываем элемент

if(x==0)printf("\nСписок пуст");//если введен 0

else{

struct list \*head = (struct list\*)malloc(sizeof(struct list));//выделение памяти под корень

head=init(x);//корень

struct list \*uzel = (struct list\*)malloc(sizeof(struct list));//выделение памяти под узел

uzel = head;//узел

while(x!=0){//пока не введен 0

scanf("%d",&x);//считываем элемент

if(x!=0){

n++;//количество элементов

uzel=addelem(uzel,x);//добавляем узел

}

}

addend(head,n,uzel);//добавляем конечный элемент

uzel = head;//переходим в начало

printf("Начальное место: %d", uzel->key);

while(i!=0){//пока не введен 0

printf("\nВыберете команду: 1-сдвинуться вправо,2-перейти в начало,3-перейти вниз,0-завершить\n");

scanf("%d",&i);//считываем команду

if(i==1){//сдвиг вправо

if(uzel->next !=NULL) uzel=uzel->next;//передвигаемся по списку если не конец

else{ //пути нет

printf("\nЗначение->%d",uzel->key);

printf("\nПути нет");

}

}

else if(i==2) uzel=head;//переход в начало списка;

else{ //переход вниз

if(uzel->end != NULL) uzel = uzel->end;

}

if(i!=0)printf("\nЗначение->%d",uzel->key);//вывод значеня списка

}

}

}