

**Университет ИТМО, кафедра ВТ**

**Лабораторная работа №4 по  
Языкам Системного Программирования**

Работу выполнил  
студент группы Р3200

**Рогов Я. С.**

Преподаватели:

**Жирков И.О.**

**Балакшин П.В.**

Санкт-Петербург, 2016

**Задание:** реализовать связный список и базовые функции работы с ним, а также написать программу, которая принимает числа из потока ввода, сохраняет их в обратном порядке в связном списке и, после окончания ввода (символом EOF), выводит их сумму.

## llist.h

```
#ifndef _LLIST_H_
#define _LLIST_H_

typedef struct llist_t{
    /* next = NULL means this element is the
    last */
    struct llist_t *next;
    int value;
} llist_t;

void llist_add(int value, llist_t **phead);

/* (list) -> (new_element llist) */
void llist_add_front(int value, llist_t **phead);

/* (list) -> (list new_element) */
void llist_add_back(int value, llist_t **phead);

void llist_connect(llist_t **phead1, llist_t
**phead2);

llist_t* llist_create(int value);

/* free llist's memory */
void llist_free(llist_t **phead);

/* get value at index */
int llist_get(int index, llist_t **phead);

void llist_insert(int index, int value, llist_t
**phead);

size_t llist_length(llist_t **phead);

/* get pointer to llist element at index */
llist_t* llist_node_at(int index, llist_t
**phead);

/* get llist's values sum */
int llist_sum(llist_t **phead);

#endif /* _LLIST_H_ */
```

## llist.c

```
#include <stdlib.h>
#include "llist.h"

void llist_add(int value, llist_t **phead){
    llist_t *lp = *phead;
    llist_t *el = llist_create(value);
    while(lp->next)
        lp = lp->next;
    lp->next = el;
}

void llist_add_front(int value, llist_t **phead){
    llist_insert(0, value, phead);
}

void llist_add_back(int value, llist_t** plist){
    llist_add(value, plist);
}

void llist_connect(llist_t **phead1, llist_t
**phead2){
    llist_t *lp = llist_node_at(-1, phead1);
    lp->next = *phead2;
}

llist_t* llist_create(int value){
    llist_t *el = malloc(sizeof(llist_t));
```

```
    if(el)
        el->value = value;
        el->next = NULL;
    return el;
}

void llist_free(llist_t **phead){
    llist_t *temp;
    do{
        temp = (*phead)->next;
        free(*phead);

        /* end: *phead = last_elem->next = NULL */
        *phead = temp;
    } while(*phead);
}

int llist_get(int index, llist_t **phead){
    llist_t *lp = llist_node_at(index, phead);
    return lp->value;
}

void llist_insert(int index, int value, llist_t
**phead){
    llist_t *prevlp;
    llist_t *lp;

    lp = llist_create(value);

    /* if insertion in the head */
    if (index==0){
        prevlp = *phead;
        lp->next = prevlp;
        *phead = lp;
    }
    else{
        prevlp = llist_node_at(index-1,
phead);
        lp->next = prevlp->next;
        prevlp->next=lp;
    }
}

size_t llist_length(llist_t **phead){
    llist_t *lp = *phead;
    size_t result = 1;
    while(lp->next){
        lp = lp->next;
        result++;
    }
    return result;
}

llist_t* llist_node_at(int index, llist_t
**phead){
    llist_t *lp = *phead;
    if(index<0)
        index+=llist_length(phead);
    for(; index>0 && lp->next; index--, lp =
lp->next);
    if(index)
        return NULL;
    return lp;
}

int llist_sum(llist_t **phead){
    llist_t *lp = *phead;
    int result = 0;
    while(lp) {
        result += lp->value;
        lp = lp->next;
    }
    return result;
}
```

## program.c

```
#include <stdio.h>
#include <stdlib.h>
#include "llist.h"

static const char* const hello = "Your numbers, please:\n";
static const char* const urbad = "Correct number, please.\n";

static char inputnumber(int* num){
    static char buffer[128];
    char * endp = buffer;
    char endc;

    while(1){
        endc = scanf("%s", buffer);
        *num = strtol(buffer, &endp, 10);

        /* if buffer parsed and ends correctly */
        if(endp!=buffer && *endp==0)
            break;
        else if(EOF==endc)
            return endc;

        printf(urbad);
    }
    return endc;
}

int main(){
    llist_t *llist;
    int num;

    printf(hello);

    if(EOF==inputnumber(&num)){
        printf("\nYou're laconic\n");
        return 1;
    }
    llist = llist_create(num);

    while(EOF != inputnumber(&num))
        llist_add_front(num, &llist);
    printf("List's sum: %d\n", llist_sum(&llist));

    llist_free(&llist);
    return 0;
}
```

**Вывод:** в ходе выполнения данной лабораторной работы я ознакомился с использованием структур в С, изучил стандарты именования типов, переменных, функций, а также научился переиспользовать код в программах.