

Отчет по лабораторной работе № VIII по курсу «Практикум на ЭВМ»

Студент группы М8О-105Б-21 Минеева Светлана Алексеевна, № по списку 14

Контакты e-mail: svetlana.mineewa2003@yandex.ru

Работа выполнена: «26» мая 2022 г.

Преподаватель: Вячеслав Константинович Титов каф. 805

Отчет сдан «26» мая 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Линейные списки.

- 2. Цель работы:** Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры. Навигацию по списку следует реализовать с применением итераторов. Предусмотреть выполнение одного нестандартного и четырех стандартных действий:
1. Печать списка.
 2. Вставка нового элемента в список.
 3. Удаление элемента из списка.
 4. Подсчет длины списка.

- 3. Задание (вариант № 2, 2, 15):** Тип элемента списка: 2. Вещественный.
Вид списка: 2. Линейный однонаправленный.
Нестандартное действие: 15. Переставить первую и вторую половины списка.

- 4. Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор 2,9 GHz 2-ядерный процессор Intel Core i5 с ОП 8 Гб, НМД 500 Гб. Монитор 13,3-дюймовый (2560 x 1600).

Другие устройства _____

5. Программное обеспечение (лабораторное):

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Terminal версия 2.10

интерпретатор команд bash версия 3.2.

Система программирования _____ версия _____

Редактор текстов emacs версия 27.2-2

Утилиты операционной системы cat, ls, g++ и другие

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Алгоритм работы:

Программа реализует 11 действий. Рассмотрим каждый подробнее:

- 0) *Выход из программы.*
- 1) *Генерация списка.* Получаем на ввод количество элементов списка, а после создаем список с данным количеством элементов, значения которых даются рандомно, добавление каждого нового элемента происходит в конец списка.
- 2) *Добавление элемента в конец списка.* На ввод получаем значение добавляемого элемента, проверяем наш список на пустоту. Если он пуст, то мы добавляем в него первый элемент, указатели tail и list сохраняют данную позицию, а указатель на следующий элемент показывает на пустой блок. Если он не пуст, то используя указатель на последний элемент списка tail, мы создаем новый блок, следующий за ним, даем ему введенное значение, указатель на следующий элемент показывает на пустой блок, а также данный блок фиксируется за tail.
- 3) *Добавление элемента в начало списка.* На ввод получаем значение добавляемого элемента, проверяем наш список на пустоту. Если он пуст, то мы добавляем в него первый элемент, указатель tail сохраняет данную позицию, а указатель на следующий элемент показывает на пустой блок. Если он не пуст, то создаем новый блок, который принимает введенное значение, как на следующий элемент он указывает на начало списка, после указатель на начало списка list принимает адрес нового блока.
- 4) *Удаление элемента списка.* На ввод получаем значение элемента, который нужно удалить. Проверяем не первый ли это элемент списка, если да, то смотрим есть ли еще в списке элементы. Если больше элементов нет, то список получает значение NULL. Если еще есть элементы, то указатель на первый элемент list становится равным следующему за ним элементу, а прошлый первый элемент удаляется. Если первый элемент не равен введенному значению, то проходим по списку, и найдя нужный элемент удаляем его, заранее связав предыдущий элемент со следующим за ним элементом. Если после прохода по списку, элемент с введенным значением не был найден, выводим данный факт.
- 5) *Печать списка.* Двигаемся по списку, пока блок не равен NULL, и выводим на экран значение элемента.
- 6) *Вставка элемента в список.* На ввод получаем значение элемента, который нужно вставить, и значение элемента, за которым следует вставить новый элемент. Проходим по списку и находим элемент со значением, за которым нужно вставить новый элемент. Создаем новый блок, который принимает значение нового элемента, его связываем указателем next со следующим элементом за тем, за которым его вставляем. Элемент же, за которым вставляем новый элемент, связывается указателем next с новым элементом.
- 7) *Удаление списка.* Двигаемся по списку и удаляем каждый его элемент, указатель на первый элемент list становится равным нулю.
- 8) *Длина списка.* Двигаемся по списку, пока блок не равен NULL, считаем количество элементов и возвращаем это значение.
- 9) *Вывод меню.*
- 10) *Перестановка первой и второй половины списка.* Мы должны переставить местами первую и вторую половину списка, однако, если существует срединный элемент, то оставляем его на своем месте. Считаем длину имеющегося списка, равную n, и n/2 раз мы берем i-ый ($i=0,1,\dots,n/2-1$) элемент первой половины списка b и меняем его местами с i-ым элементом второй половины списка (если n-четное, то во всем массиве он будет $n/2+i$, иначе - $n/2+i+1$).

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
```

```
struct ls;
typedef ls * link;
```

```
struct ls{
    float body;
    link next;}
*list, *t, *f,*tail;
```

```
int i,n,k,c;
float m,m1;
```

```

void printlist(){
    if(!list) printf("\nList is empty!\n");
    else {printf("\nList:\n[ "); t=list;
    do { printf("%2.2f ",t->body); t=t->next;}
        while(t!=NULL); printf("\n");}}

void add(float m){
    if(!list){ list=new ls; list->body=m; list->next=NULL; tail=list;}
    else { t=new ls; t->body=m; t->next=NULL; tail->next=t; tail=t;}}

void addfirst(float m){
    if (!list){ list = new ls; list->body = m; list->next = NULL; tail=list;}
    else { t = new ls; t->body = m; tail->next = NULL; t->next = list; list = t;}}

void genlist(int n){
    for(i=0;i<n;i++){
        m=(float)(rand()%10000)/100; add(m); }}

void deleting(float m){
    if (list->body == m){
        f = list;
        if (list->next == NULL){
            list = NULL;}
        else {
            list = list->next;}
        delete f;}
    else{
        t = list; c = 0;
        do {
            if (t->next->body == m){
                f = t->next; t->next = f->next; delete f; c = 1; break;}
            t = t->next;}
        while (t->next != NULL); if (!c){
            printf("\nItem with this value not exists!\n");}} }

void insert(float m, float m1){
    t = list; c = 0;
    if (t->body == m){ f = new ls; f->body = m1; f->next = t->next; t->next = f; return;}
    do{
        if (t->next->body == m){
            f = new ls; f->body = m1;
            f->next = t->next->next; t->next->next = f; c = 1; break;}
        t = t->next;}
    while (t != tail);
    if (!c){
        printf("\nItem with this value not exists!\n"); return;}}

int size(){
    t=list; n=0; if(t)do{t=t->next; n++;} while(t!=NULL); return n;}

void action(int n){
    float a;
    for (int j = 0; j < n/2; j++){ t=list;
    for (int i = 0; i < j; i++) {t = t->next;} //printf(" %d",i);}
    a = t->body; ls *t1 = t;
    if (n%2==0){ for (int k = j; k < n/2+j; k++) {t = t->next; }}
    else{ for (int k = j; k < n/2+j+1; k++) {t = t->next; }}
    t1->body = t->body;
    t->body = a; }}

int main(){
    list=0; int n, k=9; float m,m1;
    for(;;){
        if(k==9) printf("\nInput from 1 to 10 or 0 for actions:\n"
            "1. Generation list.\n2. Addition new items in end of list.\n"

```

```

"3. Addition new items in begin of list.\n"
"4. Deleting from list.\n5. Print list.\n"
    "6. Inserting in list.\n7. Erase list.\n"
    "8. Length of list.\n9. Print this MENU\n"
    "10. Main Action\n"
"0. Exit out program.\n");
    printf("\nInput number of MENU: k = "); scanf("%d",&k);
if(!k) break;
if(k==1) {printf("\nInput number of new items of list: n = "); scanf("%d",&n);
    genlist(n);}
else if(k==2) { printf("\nInput value of adding item: m = "); scanf("%f",&m);
    add(m);}
else if(k==3) { printf("\nInput value of adding item: m = "); scanf("%f",&m);
    addfirst(m);}
else if(k==4) { if(!list) { printf("\nList is empty!\n"); continue;}
    printf("\nInput value of deleting item: m = "); scanf("%f",&m);
    deleting(m);}
else if(k==5) if(!list)printf("\nList is empty!\n");else
    printlist();
else if(k==6) {if(!list) { printf("\nList is empty!\n"); continue;}
    printf("\nInput value inserting item: m1 = "); scanf("%f",&m1);
    printf("\nInput value item after which to insert: m = "); scanf("%f",&m);
    insert(m,m1);}
else if(k==7) { t=list; if(t)do {f=t; t=t->next; delete f;} while(t!=NULL); list=0;}
else if(k==8) { int f=size(); printf("\nLength of list = %d\n", f);}
else if(k==10){ if(!list) { printf("\nList is empty!\n"); continue;}
    int f=size(); action(f);}
else if(k>10) printf("\nHave no such number of MENU\n");} }

```

Так как список заполняется случайными значениями, то пункт «тесты» создается уже после пункта «распечатка протокола».

Тесты:

1) [12.90 52.49 0.73 77.77 36.58 89.30 45.67]

Вывод: [36.58 89.30 45.67 77.77 12.90 52.49 0.73]

2) [12.72 75.44 8.78 79.23 77.09 44.40]

Вывод: [79.23 77.09 44.40 12.72 75.44 8.78]

3) [81.65 44.92 30.42 79.87 25.03 23.27 17.29 88.40 26.12 43.03 31.69]

Вывод: [17.29 88.40 26.12 43.03 31.69 23.27 81.65 44.92 30.42 79.87 25.03]

4) [77.09 71.57 95.60 9.33]

Вывод: [95.60 9.33 77.09 71.57]

5) [30.99 2.78]

Вывод: [2.78 30.99]

6) [18.16]

Вывод: [18.16]

7) []

Вывод: List is empty!

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

Last login: Wed May 25 20:30:09 on ttys000

The default interactive shell is now zsh.

To update your account to use zsh, please run `chsh -s /bin/zsh`.

For more details, please visit <https://support.apple.com/kb/HT208050>.

MacBook-Pro-MacBook:~ macbookpro\$ cat zag.txt

* Минеева Светлана Алексеевна *

* M80-105Б-21 *

* Лабораторная работа №VIII *

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -4

kd.py.save

lab8.cpp

pslog_20200707_123036.log

zag.txt

MacBook-Pro-MacBook:~ macbookpro\$ cat lab8.cpp

```
#include<stdio.h>
```

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

```
#include<string.h>
```

```
#include<time.h>
```

```
struct ls;
```

```
typedef ls * link;
```

```
struct ls{
```

```
    float body;
```

```
    link next;}
```

```
*list, *t, *f,*tail;
```

```
int i,n,k,c;
```

```
float m,m1;
```

```
void printlist(){
```

```
    if(!list) printf("\nList is empty!\n");
```

```
    else {printf("\nList:\n[ "); t=list;
```

```
    do { printf("%2.2f ",t->body); t=t->next;}
```

```
        while(t!=NULL); printf("]\n");}}
```

```
void add(float m){
```

```
    if(!list){ list=new ls; list->body=m; list->next=NULL; tail=list;}
```

```
    else { t=new ls; t->body=m; t->next=NULL; tail->next=t; tail=t;}}
```

```
void addfirst(float m){
```

```
    if (!list){ list = new ls; list->body = m; list->next = NULL; tail=list;}
```

```
    else { t = new ls; t->body = m; tail->next = NULL; t->next = list; list = t;}}
```

```
void genlist(int n){
```

```
    for(i=0;i<n;i++){
```

```
        m=(float)(rand()%10000)/100; add(m); }}
```

```
void deleting(float m){
```

```
    if (list->body == m){
```

```
        f = list;
```

```
        if (list->next == NULL){
```

```
            list = NULL;}
```

```
        else {
```

```
            list = list->next;}
```

```
        delete f;}
```

```
    else{
```

```
        t = list; c = 0;
```

```
        do {
```

```
            if (t->next->body == m){
```

```
                f = t->next; t->next = f->next; delete f; c = 1; break;}
```

```

        t = t->next;}
    while (t->next != NULL); if (!c){
        printf("\nItem with this value not exists!\n");}}}

void insert(float m, float m1){
    t = list; c = 0;
    if (t->body == m){ f = new ls; f->body = m1; f->next = t->next; t->next = f;
return;}
    do{
        if (t->next->body == m){
            f = new ls; f->body = m1;
            f->next = t->next->next; t->next->next = f; c = 1; break;}
        t = t->next;}
    while (t != tail);
    if (!c){
        printf("\nItem with this value not exists!\n"); return;}}

int size(){
    t=list; n=0; if(t)do{t=t->next; n++;} while(t!=NULL); return n;}

void action(int n){
    float a;
    for (int j = 0; j < n/2; j++){ t=list;
    for (int i = 0; i < j; i++) {t = t->next;} //printf(" %d",i);}
    a = t->body; ls *t1 = t;
    if (n%2==0){ for (int k = j; k < n/2+j; k++) {t = t->next; }}
    else{ for (int k = j; k < n/2+j+1; k++) {t = t->next; }}
    t1->body = t->body;
    t->body = a; }}

int main(){
    list=0; int n, k=9; float m,m1;
    for(;;){
        if(k==9) printf("\nInput from 1 to 10 or 0 for actions:\n"
            "1. Generation list.\n2. Addition new items in end of list.\n"
            "3. Addition new items in begin of list.\n"
            "4. Deleting from list.\n5. Print list.\n"
            "6. Inserting in list.\n7. Erase list.\n"
            "8. Length of list.\n9. Print this MENU\n"
            "10. Main Action\n"
            "0. Exit out program.\n");
        printf("\nInput number of MENU: k = "); scanf("%d",&k);
        if(!k) break;
        if(k==1) {printf("\nInput number of new items of list: n = "); scanf("%d",&n);
            genlist(n);}
        else if(k==2) { printf("\nInput value of adding item: m = "); scanf("%f",&m);
            add(m);}
        else if(k==3) { printf("\nInput value of adding item: m = "); scanf("%f",&m);
            addfirst(m);}
        else if(k==4) { if(!list) { printf("\nList is empty!\n"); continue;}
            printf("\nInput value of deleting item: m = "); scanf("%f",&m);
            deleting(m);}
        else if(k==5) if(!list)printf("\nList is empty!\n");else
            printlist();
        else if(k==6) {if(!list) { printf("\nList is empty!\n"); continue;}
            printf("\nInput value inserting item: m1 = "); scanf("%f",&m1);
            printf("\nInput value item after which to insert: m = ");
scanf("%f",&m);
            insert(m,m1);}
        else if(k==7) { t=list; if(t)do{f=t; t=t->next; delete f;} while(t!=NULL);
list=0;}
        else if(k==8) { int f=size(); printf("\nLength of list = %d\n", f);}
        else if(k==10){ if(!list) { printf("\nList is empty!\n"); continue;}
            int f=size(); action(f);}
        else if(k>10) printf("\nHave no such number of MENU\n");}}

```

MacBook-Pro-MacBook:~ macbookpro\$ g++ -Wall -o lab8 lab8.cpp

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -4

```
lab8
lab8.cpp
pslog_20200707_123036.log
zag.txt
MacBook-Pro-MacBook:~ macbookpro$ ./lab8
```

Input from 1 to 10 or 0 for actions:

1. Generation list.
2. Addition new items in end of list.
3. Addition new items in begin of list.
4. Deleting from list.
5. Print list.
6. Inserting in list.
7. Erase list.
8. Length of list.
9. Print this MENU
10. Main Action
0. Exit out program.

Input number of MENU: k = 1

Input number of new items of list: n = 5

Input number of MENU: k = 5

List:
[68.07 52.49 0.73 36.58 89.30]

Input number of MENU: k = 8

Length of list = 5

Input number of MENU: k = 2

Input value of adding item: m = 45.67

Input number of MENU: k = 5

List:
[68.07 52.49 0.73 36.58 89.30 45.67]

Input number of MENU: k = 3

Input value of adding item: m = 12.9

Input number of MENU: k = 5

List:
[12.90 68.07 52.49 0.73 36.58 89.30 45.67]

Input number of MENU: k = 8

Length of list = 7

Input number of MENU: k = 4

Input value of deleting item: m = 68.07

Input number of MENU: k = 5

List:
[12.90 52.49 0.73 36.58 89.30 45.67]

Input number of MENU: k = 6

Input value inserting item: m1 = 77.77

Input value item after which to insert: m = 0.73

Input number of MENU: k = 5

List:
[12.90 52.49 0.73 77.77 36.58 89.30 45.67]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:
[36.58 89.30 45.67 77.77 12.90 52.49 0.73]

Input number of MENU: k = 9

Input from 1 to 10 or 0 for actions:
1. Generation list.
2. Addition new items in end of list.
3. Addition new items in begin of list.
4. Deleting from list.
5. Print list.
6. Inserting in list.
7. Erase list.
8. Length of list.
9. Print this MENU
10. Main Action
0. Exit out program.

Input number of MENU: k = 7

Input number of MENU: k = 5

List is empty!

Input number of MENU: k = 1

Input number of new items of list: n = 6

Input number of MENU: k = 5

List:
[12.72 75.44 8.78 79.23 77.09 44.40]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:
[79.23 77.09 44.40 12.72 75.44 8.78]

Input number of MENU: k = 7

Input number of MENU: k = 1

Input number of new items of list: n = 11

Input number of MENU: k = 5

List:
[81.65 44.92 30.42 79.87 25.03 23.27 17.29 88.40 26.12 43.03 31.69]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:

[17.29 88.40 26.12 43.03 31.69 23.27 81.65 44.92 30.42 79.87 25.03]

Input number of MENU: k = 7

Input number of MENU: k = 5

List is empty!

Input number of MENU: k = 1

Input number of new items of list: n = 4

Input number of MENU: k = 5

List:
[77.09 71.57 95.60 9.33]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:
[95.60 9.33 77.09 71.57]

Input number of MENU: k = 7

Input number of MENU: k = 1

Input number of new items of list: n = 2

Input number of MENU: k = 5

List:
[30.99 2.78]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:
[2.78 30.99]

Input number of MENU: k = 7

Input number of MENU: k = 1

Input number of new items of list: n = 1

Input number of MENU: k = 5

List:
[18.16]

Input number of MENU: k = 10

Input number of MENU: k = 5

List:
[18.16]

Input number of MENU: k = 7

Input number of MENU: k = 5

List is empty!

Input number of MENU: k = 10

List is empty!

Input number of MENU: k = 0

MacBook-Pro-MacBook:~ macbookpro\$

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	Дом.	25.05.22	20:00	Неверная работа оригинального действия	Ошибка исправлена; введено ответвление для нечетной длины	Если длина списка нечетная, то мы должны пропускать в обменном процессе серединный элемент

10. Замечания автора по существу работы

Эта лабораторная работа очень полезна, она отлично развивает мышление и учит программированию на языке Си.

11. Выводы

Я составила и отладила программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры.

Недочёты при выполнении задания могут быть устранены следующим образом: больше практиковаться в написании программ на языке Си.

Подпись студента Минеева С.А