

Отчет по Курсовой работе № 8

по курсу: 1 фундаментальная информатика

студент группы : М8О-105Б-21 Титеев Рамиль Маратович , № по списку: 23

Адреса www, e-mail, jabber, skype: derol.gym@gmail.com

Работа выполнена: "17 апреля 2022г"

Преподаватель: каф. 806 В.К.Титов

Входной контроль знаний с оценкой: _____

Отчет сдан " __ " _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Линейные списки.

2. Цель работы: Составить и отладить программу на языке Си обработки линейного списка заданной организации с отображением списка на динамические структуры.

3. Задание: Вид списка: линейный двунаправленный;

Нестандартное действие: исключить из списка последние k элементов.

Если в списке менее k элементов, то не менять его

4. Оборудование(лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ ГБ

НМД _____ ГБ. Терминал _____ адрес _____. Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 4600 @ 6x 3.0 GHz , ОП 16384 МБ, НМД _____ ГБ. Монитор Встроенный

Другие устройства _____

5. Программное обеспечение(лабораторное):

Операционная система семейства UNIX, наименование _____ версия _____

Интерпретатор команд: _____ версия _____

Система программирования: _____ версия _____

Редактор текстов: _____ версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 20.04

Интерпретатор команд: bash версия _____

Система программирования: C версия _____

Редактор текстов: Emacs версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: /usr/bin , а также /bin

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

При запуске программы вызывается меню из 11 функций.

0: программа останавливает работу;

1: генерируется список из элементов, число которых вводится в консоли. Генерация происходит следующим образом: в цикле n раз создается новый элемент со случайным значением, указатель на следующий элемент последнего элемента списка указывает на новый элемент, указатель нового элемента на следующий указывает на первый элемент списка;

2: добавляется новый элемент в конец списка. Действия аналогичны генерации списка, только значение принимается из консоли и выполняется это действие один раз;

3: добавляет элемент в начало списка; Происходит это следующим образом: создается новый элемент со значением, введенном из консоли. Указатель нового элемента на следующий указывает на первый элемент прошлого списка, а указатель на предыдущий элемент указывает на последний элемент прошлого списка.

4: Удаляет из списка введенный элемент. Для этого циклом проходимся по списку, пока не встретим нужное значение, а затем с помощью указателей убираем его из списка, и потом удаляем данный элемент из памяти;

5: Выводит все элементы списка. Циклом проходим все элементы списка и выводим их значение;

6: Вставляет элемент после указанного. Происходит это следующим образом: циклом идем по списку, пока не встретим элемент, имеющий нужное значение, затем создаем новый элемент, указатель на следующий элемент у нового элемента указывает на элемент, который ранее шел после найденного, указатель на следующий элемент у найденного ранее элемента указываем на только что созданный элемент;

7: очищает список. Циклом проходим по списку пока не дойдем до конца и удаляем из памяти каждый элемент;

8: выводит длину списка. Циклом проходим по списку пока не дойдем до конца и считаем кол-во элементов;

9: выводит меню повторно;

10: вызывается особое действие над списком. Требуется ввести кол-во удаляемых элементов. Если число больше длины списка, то ничего не происходит и выводится ошибка. Если же оно меньше или равно, то в цикле который повторяется столько раз, сколько было введено, обращаемся к предпоследнему элементу через указатели, указатель на следующий элемент выбранного элемента указываем на начало списка и удаляем из памяти последний элемент списка.

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

curs_8.cpp:

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

struct ls;
typedef ls * link;
typedef enum{Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, UNK} item;
char month[12][4]={"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                  "Oct", "Nov", "Dec"};

struct ls{
    item body;
    link pre, next;
};

int i, n, k, c;
item m, m1;
ls *list, *t, *f;

void randomize(){
```

```

    long a=time(0);
    srand(a);
}

void add(item m){
    if(!list){
        list = new ls;
        list->body = m;
        list->next = list;
        list->pre = list;
    }
    else{
        t = new ls;
        t->body = m;
        f = list->pre;
        list->pre = t;
        f->next = t;
        t->next = list;
        t->pre = f;
    }
}

void addfirst(item m){
    if(!list){
        list = new ls;
        list->body = m;
        list->next = list;
        list->pre = list;
    }
    else{
        t = new ls;
        t->body = m;
        f = list;
        list = t;
        list->next = f;
        list->pre = f->pre;
        f->pre->next = list;
        f->pre = list;
    }
}

void genlist(int n){
    for(i=0;i<n;i++){
        m=item(rand()%12);
        add(m);
    }
}

void printlist(){
    if(!list) printf("\nList is empty!\n");
    else {
        printf("\nList:\n[ ");
        t=list;
        do{

```

```

        printf("%s ",month[t->body]);
        t=t->next;
    } while(t!=list);
    printf("]\n");
}

}

item read_val(){
    char b[4];
    scanf("%s",b);
    for(i=0;i<12;i++){
        if(!strcmp(b,month[i])){
            return item(i);
        }
    }
    return UNK;
}

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

void insert(item m, item m1){
    t=list;
    c=0;
    do{
        if(t->body==m){
            f=new ls;
            f->body=m1;

```

```

        f->next=t->next;
        f->pre=t;
        t->next->pre=f;
        t->next=f;
        c=1;
        break;
    }
    t=t->next;
}
while(t!=list);
if(!c) printf("\nItem with this value not exists!\n");
}

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

void action(int q){
    t=list;
    if (length() < q){
        printf("Can`t do this!!!");
        return;
    }
    else{
        int c = q;
        while(c>0){
            f = t->pre;
            f->pre->next = t;
            t->pre = f->pre;
            c--;
            delete f;
        };
    }
}

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

```

```

    "3. Addition new items in begin of list.\n"
    "4. Deleting from list.\n"
    "5. Print list.\n"
    "6. Inserting in list.\n"
    "7. Erase list.\n"
        "8. Lenght of list.\n"
    "9. Print this MENU\n"
        "10. Main Action\n"
    "0. Exit out program.\n");}
printf("\nInput number of MENU: "); 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 = ");
    m=read_val();
    add(m);
}
else if(k==3){
    printf("\nInput value of adding item: m = ");
    m=read_val();
    addfirst(m);
}
else if(k==4){
    if(!list){
        printf("\nList is empty!\n");
        continue;
    }
    printf("\nInput value of deleting item: m = ");
    m=read_val(); deleting(m);
}
else if(k==5) printlist();
else if(k==6){
    if(!list){
        printf("\nList is empty!\n");
        continue;
    }
    printf("\nInput value inserting item: m1 = ");
    m1=read_val();
    printf("\nInput value item after which to insert: m = ");
    m=read_val();
    insert(m,m1);
}
else if(k==7){
    t=list;
    if(t){
        do{
            f=t;
            t=t->next;
            delete f;
        }while(t!=list);
    }
}

```

```

        list=0;
    }
}
else if(k==8){
    printf("\nLenght of list = %d\n", length());
}
else if(k==10){
    if(!list){
        printf("\nList is empty!\n");
        continue;
    }
    int q;
    printf("\nInput value for main action: k=");
    scanf("%d", &q);
    action(q);
}
else if(k>10) printf("Have no such number of MENU\n");
}
}

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```

constantfear@constantfear:~/projects/laboratory/2_semester/curs_8$ cat header.txt
*****
*                               *
*           Курсовая работа №8   *
*           Линейные списки.     *
*           Выполнил студент гр. М80-105-Б   *
*           Тигеев Рамиль Маратович   *
*                               *
*****
constantfear@constantfear:~/projects/laboratory/2_semester/curs_8$ g++ curs_8.cpp
constantfear@constantfear:~/projects/laboratory/2_semester/curs_8$ ./a.out

```

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. Lenght of list.
9. Print this MENU
10. Main Action
0. Exit out program.

Input number of MENU: 1

Input number of new items of list: n=10

Input number of MENU: 5

List:

[Jan Jul Jun Nov Aug Jan Sep Dec Nov Dec]

Input number of MENU: 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. Lenght of list.
9. Print this MENU
10. Main Action
0. Exit out program.

Input number of MENU: 2

Input value of adding item: m = Jun

Input number of MENU: 5

List:

[Jan Jul Jun Nov Aug Jan Sep Dec Nov Dec Jun]

Input number of MENU: 8

Lenght of list = 11

Input number of MENU: 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. Lenght of list.
9. Print this MENU
10. Main Action
0. Exit out program.

Input number of MENU: 10

Input value for main action: k=4

Input number of MENU: 5

List:

[Jan Jul Jun Nov Aug Jan Sep]

Input number of MENU: 10

Input value for main action: k=10

Can`t do this!!!

Input number of MENU: 0

9. Дневник отладки должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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

10. Замечание автора по существу работы _____

11. Выводы _____ Я научился работать с линейными списками.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____