**Отчет по курсовой работе №8** по курсу\_1\_\_\_\_\_

|  |  |
| --- | --- |
|  | Студент группы М80-105Б-21 Жилов Андрей, № по списку 3  Контакты www, e-mail: [klzxrcn3692@outlook.com](mailto:klzxrcn3692@outlook.com)\_\_\_\_\_\_\_\_\_\_\_  Работа выполнена: «24» апреля 2021г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Преподаватель: В. К. Титов каф. 806\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отчет сдан «24» апреля 2021 г., итоговая оценка \_\_\_\_\_\_\_\_\_  Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. **Тема:** Линейные списки
2. **Цель работы:** Составить программу на языке Си обработки линейного списка заданной организации с отображением списка на динамические структуры.
3. **Задание:** (*Вариант №3*):

Вид списка: линейный однонаправленный

Нестандартное действие: обменять местами (k − 1) -й и (k + 1) -й элементы списка (k задается в качестве параметра);

1. **Оборудование(лабораторное)**:

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

НМД - ГБ, терминал- адрес -, принтер –

Другие устройства –

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 3500U, с ОП 8 ГБ, НМД 256 ГБ. Монитор 1920x1080 пикс.

Другие устройства –

1. **Программное обеспечение:**

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 GNU/Linux, наименование Kali версия 5.10.0

интерпретатор команд Bash версия 5.1.8(1)-release

Система программирования \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_версия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

редактор текстов emacs версия 27.1

Утилиты операционной системы cmp, diff, tail, od, vi, vim, ed, head, du, xargs, indent, comm, grep, tee, sum, mc, awk, iconv, file,df,split, wc, join, tr,cut,tar,sed,patch,touch, paste, mktemp, dd,sort, uniq, nroff, gzip, bzip2, md5, find, cpp

Прикладные системы и программы –

Местонахождение и имена файлов программ и данных -

1. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок,

При запуске программы вызывается меню из 10 функций.

Вводится цифра от 0 до 9, которая вызывает определенную функцию:

0: программа останавливает работу;  
1: генерируется список из элементов, количество которых вводится в консоли. Генерация происходит следующим образом: в цикле n раз создается новый элемент со случайным значением, указатель прошлого элемента указывает на новый, указатель нового элемента указывает нуль;  
2: добавляется новый элемент в конец списка. Действия аналогичны генерации списка, только значение принимается из консоли;  
3: добавляет элемент в начало списка. Создается новый элемент, который принимает значение первого элемента, и указывает на элемент, следующий после первого, в свою очередь первый элемент принимает введенное значение и указывает на новый элемент.  
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: Вставляет элемент после указанного. Циклом идем по списку, пока не встретим элемент, имеющий нужное значение, затем создаем новый элемент, который принимает введённое значение и указывает на следующий элемент, предыдущий элемент в свою очередь указывает на только, что вставленный.

6: Выводит все элементы списка. Циклом проходим все элементы списка и выводим их значение;  
7: очищает список. Циклом проходим по списку пока не дойдем до конца и удаляем из памяти каждый  
элемент;  
8: выводит меню повторно;  
9: вызывается особое действие над списком. Требуется ввести значение, далее проходимся циклом, пока не дойдем до элемента, соответствующему введённому номеру, затем мы меняем значения предыдущего и последующего элементов местами.

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

Curs.c:

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <string.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 next;

} \*list, \*young, \*old, \*jump, \*f;

item element;

void kryzhkas\_random()

{

    long a = time(0);

    srand(a);

}

void addlast(item i)

{

    if(!list)

    {

        old = new ls;

        list = new ls;

        list->body=i;

        list->next=NULL;

        old=list;

    }

    else

    {

        young=new ls;

        young->body=i;

        young->next=NULL;

        old->next=young;

        old=young;

    }

}

void addfirst(item i)

{

    if(!list)

    {

        list = new ls;

        list->body=i;

        list->next=NULL;

    }

    else

    {

        young = new ls;

        young->next=list->next;

        list->next=young;

        young->body=list->body;

        list->body=i;

    }

    //printf("%s\n",month[i]);

}

void generate\_list(int number)

{

    for(int i=0;i<number;++i)

    {

        element=item(rand()%12);

        addlast(element);

        // printf("%s\n",month[element]);

    }

}

void printlist()

{

    if(!list)

        printf("List is empty\n");

    else

    {

        jump=list;

        while(jump)

        {

           printf("%s\n",month[jump->body]);

           jump=jump->next;

        }

    }

}

item read\_val()

{

    char b[4];

    scanf("%s",b);

    for(int i=0;i<12;i++)

        if(!strcmp(b,month[i]))

            return item(i);

    return UNK;

}

void deleting(item m)

{

    if(list->body==m)

    {

        if(list->next==list)

            list=NULL;

        else

        {

            list=list->next;

        }

    }

    else

    {

        jump=list;

        int c=0;

        do

        {

            if(jump->next->body==m)

                {

                    f=jump->next;

                    jump->next=f->next;

                    delete f;

                    c=1;

                    break;

                }

            jump=jump->next;

        }while(jump->next);

        if(!c)

            printf("\nItem with this value not exists!\n");

        }

}

void insert(item m, item m1)

{

    jump=list;

    int c=0;

    do

    {

        if(jump->body==m)

        {

            young=new ls;

            young->body=m1;

            young->next=jump->next;

            jump->next=young;

            c=1;

            break;

       }

       jump=jump->next;

    }while(jump);

    if(!c)

        printf("\nItem with this value not exists!\n");

}

void action(int k)

{

    int r=1;

    jump=list;

    while(jump->next)

    {

        if(r==k)

        {

            element=old->body;

            old->body=jump->next->body;

            jump->next->body=element;

            break;

        }

        else

        {

            ++r;

            old=jump;

            jump=jump->next;

        }

    }

}

int main()

{

    kryzhkas\_random();

    int k=8,c;

    item n,m;

    while(k)

    {

        if(k==8)

        printf("Input from 0 to 10 actions:\n"

        "1.Generate list\n"

        "2.Add a new element to end of list\n"

        "3.Add a new element to begin of list\n"

        "4.Delete element from the list\n"

        "5.Inser element in list\n"

        "6.Print list\n"

        "7.Erase list\n"

        "8.Print this menu\n"

        "9.Main action\n"

        "0.Exit from the program\n\n"

        );

        else if(k==0)

            return 0;

        else if(k==1)

        {

            printf("Please enter lenth of list\n");

            scanf("%d",&c);

            generate\_list(c);

        }

        else if(k==2)

        {

            printf("Please enter a new element\n");

            addlast(read\_val());

        }

        else if(k==3)

        {

            printf("Please enter a new element\n");

            addfirst(read\_val());

        }

        else if(k==4)

        {

            printf("Please enter value of deleting item\n");

            deleting(read\_val());

        }

        else if(k==5)

        {

            printf("Please enter inserting item\n");

            m=read\_val();

            printf("Please enter value item after which to insert\n");

            n=read\_val();

            insert(n,m);

        }

        else if(k==6)

            printlist();

        else if(k==7)

        {

            jump=list;

            if(jump)

            do

            {

                young=jump;

                jump=jump->next;

                delete young;

            }while(jump!=list);

            list=0;

        }

        else if(k==9)

        {

            if(!list)

                {

                    printf("list is empty\n");

                    continue;

                }

            printf("Please enter k\n");

            scanf("%d",&c);

            action(c);

        }

        printf("Please choose action(8 for menu)\n");

        scanf("%d",&k);

    }

    return 0;

}

*Пункты 1-7 отчета составляются сторого до начала лабораторной работы.*

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

lain@navi:/mnt/d/mai/sem2/lab/curs8$ cat head\_curs\_8.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

| |

| Курсовая работа №8 |

| Линейные списки |

| Студента группы М8О-105Б-21 |

| Жилова Андрея |

| |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

lain@navi:/mnt/d/mai/sem2/lab/curs8$ g++ curs.c

lain@navi:/mnt/d/mai/sem2/lab/curs8$ ./a.out

Input from 0 to 10 actions:

1.Generate list

2.Add a new element to end of list

3.Add a new element to begin of list

4.Delete element from the list

5.Inser element in list

6.Print list

7.Erase list

8.Print this menu

9.Main action

0.Exit from the program

Please choose action(8 for menu)

1

Please enter lenth of list

7

Please choose action(8 for menu)

6

Jul

Jun

Oct

Feb

May

Oct

Mar

Please choose action(8 for menu)

2

Please enter a new element

Aug

Please choose action(8 for menu)

6

Jul

Jun

Oct

Feb

May

Oct

Mar

Aug

Please choose action(8 for menu)

8

Input from 0 to 10 actions:

1.Generate list

2.Add a new element to end of list

3.Add a new element to begin of list

4.Delete element from the list

5.Inser element in list

6.Print list

7.Erase list

8.Print this menu

9.Main action

0.Exit from the program

Please choose action(8 for menu)

3

Please enter a new element

Jan

Please choose action(8 for menu)

6

Jan

Jul

Jun

Oct

Feb

May

Oct

Mar

Aug

Please choose action(8 for menu)

4

Please enter value of deleting item

Oct

Please choose action(8 for menu)

6

Jan

Jul

Jun

Feb

May

Oct

Mar

Aug

Please choose action(8 for menu)

8

Input from 0 to 10 actions:

1.Generate list

2.Add a new element to end of list

3.Add a new element to begin of list

4.Delete element from the list

5.Inser element in list

6.Print list

7.Erase list

8.Print this menu

9.Main action

0.Exit from the program

Please choose action(8 for menu)

5

Please enter inserting item

Feb

Please enter value item after which to insert

Jan

Please choose action(8 for menu)

6

Jan

Feb

Jul

Jun

Feb

May

Oct

Mar

Aug

Please choose action(8 for menu)

9

Please enter k

4

Please choose action(8 for menu)

6

Jan

Feb

Feb

Jun

Jul

May

Oct

Mar

Aug

Please choose action(8 for menu)

8

Input from 0 to 10 actions:

1.Generate list

2.Add a new element to end of list

3.Add a new element to begin of list

4.Delete element from the list

5.Inser element in list

6.Print list

7.Erase list

8.Print this menu

9.Main action

0.Exit from the program

Please choose action(8 for menu)

0

1. **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. **Замечания автора по существу работы:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Выводы**:

В результате выполнения данной лабораторной работы я научился работать с линейными списками на языке Си

Недочеты при выполнении задания могут быть устранены следующим образом:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_