Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им.И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Алгоритмы и структуры данных»

Лабораторная работа №6

Тема: «Работа со списком»

Выполнил: студент группы

ПИ(б)-2-19 Улан уулу Нурдин

Проверила: Валеева А. А.

Бишкек – 2020

**Практическое задание**

**I. Ответы на контрольные вопросы**

1. **Какие есть особенности при добавлении элемента в односвязный список?**

В отличии от Stack и Queue в односвязном списке можно добавить куда угодно.

1. **Как произвести операцию добавления элемента по указанному порядковому номеру?**

Если хотим добавить в начало, то: создаем новую структурную переменную, даем ей элемент, который добавляем, и указываем ей **next** на существующий список;

Иначе если в конец: создаем две новых структурных переменных, в одну из них даем элемент, который добавляем в список, а во второй присваиваем для хода по списку, идем в конец, и последний **next** присваиваем первую переменную с новым элементом;

Иначе в произвольное место: создаем две новых структурных переменных, в одну из них даем элемент, который добавляем в список, а во второй присваиваем для хода по списку, идем **до** нужного места с счетчиком, когда дойдем, останавливаем цикл, указываем **next** на новый элемент, а его next на стоящий до него в этом индексе;

1. **Какие есть особенности при удалении элемента из односвязного списка?**

В отличии от Stack и Queue в односвязном списке можно удалить откуда угодно.

1. **Как произвести операцию удаления элемента по указанному порядковому номеру?**

Если хотим удалить из начала, то: создаем новую структурную переменную присваиваем ей указатель на первый элемент и освобождаем его;

Иначе если удаляем из конца, то: создаем две новых структурных переменных, в один из них присваиваем нынешний список, и через цикл идем в конец, попутно передавая указатель и элементы, во вторую переменную кроме последнего указателя и элемента, и освобождаем ее;

Иначе в произвольном месте: так же, как и удаление конца списка идем в нужное место, через счетчик попутно проверяя нужный индекс, после того как дойдем освобождаем элемент;

1. **В чем заключается поиск элемента в списке?**

Мы проходимся циклом по списку и ищем элементы, записывая в счетчик итерацию где лежит элемент.

1. **В чем состоит отличие линейного списка от кольцевого?**

В линейном списке последний элемент указывает на NULL, а в кольцевом списке для последнего элемента следующим является первый.

1. **В чем состоит отличие связного списка от стека, организованного в виде связного списка?**

Отличие состоит в том, что, в стеке мы имеем доступ только к последнему добавленному элементу. В списке же мы можемполучить доступ к произвольному элементу.

1. **В чем состоит отличие связного списка от очереди, организованного в виде связного списка?**

Отличие состоит в том, что, в очереди мы имеем доступ к элементам в таком же порядке что и клали. В списке же мы можем получить доступ к произвольному элементу.

1. **Перечислите достоинства и недостатки линейных односвязных списков.**

***Плюсы:***

1. Добавление и удаление ячеек осуществляется быстро.

2. Кроме добавления и удаления в конец, мы также можем добавить и удалить элемент в начале контейнера.

***Минусы:***

1. Медленное обращение к элементам, находящимся в центре.

2. Большие затраты на оперативную память.

1. **Объясните назначение выделенного заглавного звена в списках.**

В основном он нужен для проверки при добавлении или удалении, за частую его используют как счётчик количества элементов.

1. **Особенности организации кольцевого списка?**

Построение выполняется так же, как и в случае линейного однонаправленного списка с заглавным звеном, только после окончания ввода элементов кольцевого списка, в поле указателя последнего звена списка помещается адрес звена, следующего за заглавным.

1. **Какие ошибки могут быть допущены при сцеплении 2-х списков?**

* Не инициализация одного из списков;
* Не прибавить количество элементов одного списка к количеству элементов второго списка;
* Не очистить второй список после добавление его в первый, или наоборот;

1. **В чем заключаются недостатки односвязного списка?**

Недостатки связных списков вытекают из их главного свойства — последовательного доступа к данным:

* сложность прямого доступа к элементу
* на поля-указатели расходуется дополнительная память
* некоторые операции со списками медленнее, чем с массивами
* соседние элементы списка могут быть распределены в памяти не локально
* над связными списками, по сравнению с массивами, гораздо труднее производить параллельные векторные операции, такие, как вычисление суммы

**II. Выполнить упражнение**

**1. Постановка задачи**

1. Разработать алгоритм, блок схемы и программный код для работы с односвязным списком –

* создание списка;
* добавление элемента в список-
* по позиции;
* по значению:
* перед указанным элементом;
* после указанного элемента;
* удаление элемента из списка-
* по позиции;
* по значению:
* перед указанным элементом;
* после указанного элемента;
  + поиск элемента по ключу;
  + поиск позиции по значению элемента;
  + очистка списка;
  + вывод списка.

**2. Описание входных и выходных данных**

**Исходные данные:**

**list\* \_List = NULL;**

**void inition(list\*& li);**- функция инициализации;

**void push(list\*& li, int k);** - функция добавления;

**void add(list\*& li);** - функция для меню добавления;

**void output(list\* li);** - функция вывода;

**void deleLI(list\*& li);** - функция для меню удаления;

**void pop(list\*& li, int k);** - функция удаления;

**void searchByIndex(list\* li);** - функция поиска по индексу;

**void search(list\*& li);** - функция поиска по значению

**int searchByValue(list\* li, int value);** - функция поиска индекса значения;

**void clear(list\*& li);** - функция очищения очереди;

int main()



void inition(list\*& li);



void output(Queue\* \_First)



void searchByIndex(list\* li)



int searchByValue(list\* li, int value)



void search(list\*& li)  
 

void clear(list\*& li)



void push(list\*& li, int k)



void pop(list\*& li, int k)



void add(list\*& li)



void deleLI(list\*& li)



**3.Текст программы**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <iomanip>

#include "Windows.h"

using namespace std;

struct list {

int item;

list\* next;

};

int \_count = 0;

void inition(list\*& li);

void output(list\* li);

void searchByIndex(list\* li);

int searchByValue(list\* li, int value);

void search(list\*& li);

void clear(list\*& li);

void push(list\*& li, int k);

void pop(list\*& li, int k);

void add(list\*& li);

void deleLI(list\*& li);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

list\* \_List = new list;

bool f = true;

int choice;

\_List->next = 0;

inition(\_List);

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Поиск элемента по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Поиск элемента по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Очистить список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Выход |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(\_List); break;

case 2: output(\_List); break;

case 3: deleLI(\_List); break;

case 4: searchByIndex(\_List); break;

case 5: search(\_List); break;

case 6: clear(\_List); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

return 0;

}

void inition(list\*& li) {

int x;

cout << "Инициализируйте первый элемент для списка:";

cin >> x;

system("cls");

list\* \_li = new list;

\_li->item = x;

li->next = \_li;

li->next->next = NULL;

cout << "Первый элемент со значением = " << x << " инициализирован\n" << endl;

\_count++;

}

void output(list\* li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего выводить!!!" << endl;

} else {

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

}

}

void searchByIndex(list\* li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

} else {

int x, i = 1;

bool f = true;

list\* \_li = new list;

while (f) {

cout << "Введите индекс искоемого элемента:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

} else {

f = false;

\_li = li->next;

while (\_li != 0) {

if (i == x) {

cout << "Искомый элемент:" << endl;

cout << x << "| " << \_li->item << endl;

}

i++;

\_li = \_li->next;

}

}

}

}

}

int searchByValue(list\* li, int value) {

int i = 1, j, kol = 0, k = 0, x;

bool f = true;

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

kol++;

j = i;

}

i++;

\_li = \_li->next;

}

if (kol == 0) {

return 0;

} else if (kol == 1) {

return j;

} else {

while (f) {

for (i = 0; i < kol; i++) {

cout << i + 1 << "\t|" << value << endl;

}

cout << "Мы нашли " << kol << " элемента с введённым значение, выберите 1 из них:";

cin >> x;

system("cls");

if (x <= 0 || x > kol) {

cout << "Ошибка при вводе!!!" << endl;

} else {

f = false;

i = 1;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

j = i;

k++;

}

if (k == x) {

break;

}

i++;

\_li = \_li->next;

}

return j;

}

}

}

return 0;

}

void search(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

} else {

bool f1 = true;

int search, x;

while (f1) {

cout << "Введите значение элемента который ищем:";

cin >> x;

system("cls");

search = searchByValue(li, x);

if (search == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

cout << "Искомый элемент: " << endl << search << "| " << x << endl;

}

}

}

}

void clear(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего очищать!!!" << endl;

} else {

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

list\* \_LI = new list;

\_LI = \_li;

\_li = \_li->next;

delete \_LI;

}

\_count = 0;

li->next = 0;

cout << "Лист успешно очищен!!!" << endl;

}

}

void push(list\*& li, int k) {

list\* \_li = new list;

int j = 1, x;

cout << "Введите новый элемент для добавления:";

cin >> x;

system("cls");

\_li->item = x;

if (k == 1) {

\_li->next = li->next;

li->next = \_li;

} else if (k == (\_count + 1)) {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

} else {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k - 1) {

\_li->next = \_LI->next;

\_LI->next = \_li;

break;

}

j++;

\_LI = \_LI->next;

}

}

\_count++;

cout << "Элемент " << x << " успешно добавлен";

}

void pop(list\*& li, int k) {

list\* \_li = new list;

int x;

if (k == 1) {

\_li = li->next;

li->next = \_li->next;

x = \_li->item;

cout << "Элемент " << \_li->item << " успешно удален" << endl;

delete \_li;

} else if (k == \_count) {

list\* \_LI = new list;

\_LI = li;

while (\_LI->next != 0) {

\_li = \_LI;

\_LI = \_LI->next;

}

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

\_li->next = 0;

} else {

int j = 1;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k) {

\_li->next = \_LI->next;

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

break;

}

j++;

\_li = \_LI;

\_LI = \_LI->next;

}

}

\_count--;

if (\_count == 0) {

li->next = 0;

}

}

void add(list\*& li) {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню добавления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Добавление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "|3. Добавление после указанного элемента|" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс куда добавить:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, x);

cout << " в индекс = " << x << endl;

}

}

} else if (x == 2) {

if (\_count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

} else {

while (f1) {

cout << "Введите значение элемента до которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, check);

cout << " до элемента " << x << endl;

}

}

}

} else if (x == 3) {

if (\_count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

} else {

while (f1) {

cout << "Введите значение элемента после которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, (check + 1));

cout << " после элемента " << x << endl;

}

}

}

} else if (x == 4) {

f = false;

} else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

void deleLI(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего удалять!!!" << endl;

} else {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню удаления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Удаление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Удаление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление после указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Удаление по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс откуда удалить:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

pop(li, x);

}

}

} else if (x == 2) {

while (f1) {

cout << "Введите значение элемента до которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

if (check == 1) {

cout << "Нельзя удалить до " << x << " \nПотому что там ничего нет!!!" << endl;

} else {

pop(li, check - 1);

}

}

}

} else if (x == 3) {

while (f1) {

cout << "Введите значение элемента после которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

if (check == \_count) {

cout << "Нельзя удалить после " << x << " \nПотому что там ничего нет!!!" << endl;

} else {

pop(li, check + 1);

}

}

}

} else if (x == 4) {

while (f1) {

cout << "Введите значение элемента который будем удалять:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

pop(li, check);

}

}

} else if (x == 5) {

f = false;

} else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;}}}}

**1.Постановка задачи**

Разработать алгоритм, блок схему и программный код для организации кольцевого списка.

**2.Описание входных и выходных данных**

**Исходные данные:**

**void round(list\*& li, list\*& fi); -** функция зацикливания очереди;

void round(list\*& li, list\*& fi);



**3.Текст программы**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <iomanip>

#include "Windows.h"

using namespace std;

struct list {

int item;

list\* next;

};

int \_count = 0;

bool ifround = false;

void inition(list\*& li, list\*& fi);

void output(list\* li);

void searchByIndex(list\* li);

int searchByValue(list\* li, int value);

void search(list\*& li);

void clear(list\*& li);

void push(list\*& li, int k);

void pop(list\*& li, int k);

void add(list\*& li);

void deleLI(list\*& li);

void round(list\*& li, list\*& fi);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

list\* \_List = new list;

list\* \_f = new list;

bool f = true;

int choice;

int i = 1;

\_f->next = 0;

\_List->next = 0;

inition(\_List, \_f);

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Поиск элемента по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Поиск элемента по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Очистить список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Закольцевать список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 8. Выход |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(\_List); break;

case 2: output(\_List); break;

case 3: deleLI(\_List); break;

case 4: searchByIndex(\_List); break;

case 5: search(\_List); break;

case 6: clear(\_List); break;

case 7: round(\_List, \_f); break;

case 8: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

return 0;

}

void inition(list\*& li, list\*& fi) {

int x;

cout << "Инициализируйте первый элемент для списка:";

cin >> x;

system("cls");

list\* \_li = new list;

\_li->item = x;

fi = \_li;

li->next = fi;

li->next->next = NULL;

cout << "Первый элемент со значением = " << x << " инициализирован\n" << endl;

\_count++;

}

void output(list\* li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего выводить!!!" << endl;

}

else {

if (ifround == false) {

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

} else {

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0 && i < 5) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

}

}

}

void searchByIndex(list\* li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

}

else {

int x, i = 1;

bool f = true;

list\* \_li = new list;

while (f) {

cout << "Введите индекс искоемого элемента:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f = false;

\_li = li->next;

while (\_li != 0) {

if (i == x) {

cout << "Искомый элемент:" << endl;

cout << x << "| " << \_li->item << endl;

}

i++;

\_li = \_li->next;

}

}

}

}

}

int searchByValue(list\* li, int value) {

int i = 1, j, kol = 0, k = 0, x;

bool f = true;

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

kol++;

j = i;

}

i++;

\_li = \_li->next;

}

if (kol == 0) {

return 0;

}

else if (kol == 1) {

return j;

}

else {

while (f) {

for (i = 0; i < kol; i++) {

cout << i + 1 << "\t|" << value << endl;

}

cout << "Мы нашли " << kol << " элемента с введённым значение, выберите 1 из них:";

cin >> x;

system("cls");

if (x <= 0 || x > kol) {

cout << "Ошибка при вводе!!!" << endl;

}

else {

f = false;

i = 1;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

j = i;

k++;

}

if (k == x) {

break;

}

i++;

\_li = \_li->next;

}

return j;

}

}

}

return 0;

}

void search(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

}

else {

bool f1 = true;

int search, x;

while (f1) {

cout << "Введите значение элемента который ищем:";

cin >> x;

system("cls");

search = searchByValue(li, x);

if (search == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

cout << "Искомый элемент: " << endl << search << "| " << x << endl;

}

}

}

}

void clear(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего очищать!!!" << endl;

}

else {

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

list\* \_LI = new list;

\_LI = \_li;

\_li = \_li->next;

delete \_LI;

}

\_count = 0;

li->next = 0;

cout << "Лист успешно очищен!!!" << endl;

}

}

void push(list\*& li, int k) {

list\* \_li = new list;

int j = 1, x;

cout << "Введите новый элемент для добавления:";

cin >> x;

system("cls");

\_li->item = x;

if (k == 1) {

\_li->next = li->next;

li->next = \_li;

}

else if (k == (\_count + 1)) {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

}

else {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k - 1) {

\_li->next = \_LI->next;

\_LI->next = \_li;

break;

}

j++;

\_LI = \_LI->next;

}

}

\_count++;

cout << "Элемент " << x << " успешно добавлен";

}

void pop(list\*& li, int k) {

list\* \_li = new list;

int x;

if (k == 1) {

\_li = li->next;

li->next = \_li->next;

x = \_li->item;

cout << "Элемент " << \_li->item << " успешно удален" << endl;

delete \_li;

}

else if (k == \_count) {

list\* \_LI = new list;

\_LI = li;

while (\_LI->next != 0) {

\_li = \_LI;

\_LI = \_LI->next;

}

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

\_li->next = 0;

}

else {

int j = 1;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k) {

\_li->next = \_LI->next;

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

break;

}

j++;

\_li = \_LI;

\_LI = \_LI->next;

}

}

\_count--;

if (\_count == 0) {

li->next = 0;

}

}

void add(list\*& li) {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню добавления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Добавление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "|3. Добавление после указанного элемента|" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс куда добавить:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, x);

cout << " в индекс = " << x << endl;

}

}

}

else if (x == 2) {

if (\_count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

}

else {

while (f1) {

cout << "Введите значение элемента до которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, check);

cout << " до элемента " << x << endl;

}

}

}

}

else if (x == 3) {

if (\_count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

}

else {

while (f1) {

cout << "Введите значение элемента после которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, (check + 1));

cout << " после элемента " << x << endl;

}

}

}

}

else if (x == 4) {

f = false;

}

else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

void deleLI(list\*& li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего удалять!!!" << endl;

}

else {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню удаления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Удаление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Удаление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление после указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Удаление по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс откуда удалить:";

cin >> x;

system("cls");

if (x <= 0 || x > (\_count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

pop(li, x);

}

}

}

else if (x == 2) {

while (f1) {

cout << "Введите значение элемента до которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

if (check == 1) {

cout << "Нельзя удалить до " << x << " \nПотому что там ничего нет!!!" << endl;

}

else {

pop(li, check - 1);

}

}

}

}

else if (x == 3) {

while (f1) {

cout << "Введите значение элемента после которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

if (check == \_count) {

cout << "Нельзя удалить после " << x << " \nПотому что там ничего нет!!!" << endl;

}

else {

pop(li, check + 1);

}

}

}

}

else if (x == 4) {

while (f1) {

cout << "Введите значение элемента который будем удалять:";

cin >> x;

system("cls");

check = searchByValue(li, x);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

pop(li, check);

}

}

}

else if (x == 5) {

f = false;

}

else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

}

void round(list\*& li, list\*& fi) {

if (\_count <= 0) {

cout << "Лист пуст, нечего кольцевать!!!" << endl;

}

else {

if (ifround == false) {

list\* \_li = new list;

\_li = fi;

list\* \_LI = new list;

\_LI = fi;

while (\_li != 0) {

if (\_li->next == 0) {

\_li->next = \_LI;

break;

}

\_li = \_li->next;

}

ifround = true;

cout << "Список закольцован!!!" << endl;

}

else {

cout << "Список откольцован!!!" << endl;

ifround = false;

list\* \_li = new list;

\_li = fi;

for (int l = 0; l < \_count; l++) {

\_li = \_li->next;

if (l == (\_count - 2)) {

\_li->next = NULL;

}

}

}

}

}

**1.Постановка задачи**

Разработать алгоритм, блок схему и программный код для организации слияния двух, трех списков.

**2.Описание входных и выходных данных**

**Исходные данные:**

**void listnum1(list\* li, int& \_count1); -** главное меню первого списка;

**void listnum2(list\* li, int& \_count2); -** главное меню второго списка;

**void listnum3(list\* li, int& \_count3); -** главное меню третьего списка;

**void con2(list\*& li1, list\*& li2, int& count1, int& count2); -** функция для конкатенации двух списков;

**void concon2(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3); -** функция меню конкатенации двух списков;

**void con3(list\*& li1, list\*& li2, list\*& li3, int& count1, int& count2, int& count3); -** функция для конкатенации трех списков;

**void concon3(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3); -** функция меню конкатенации трех списков;

void con2(list\*& li1, list\*& li2, int& count1, int& count2);



void con3(list\*& li1, list\*& li2, list\*& li3, int& count1, int& count2, int& count3);



void concon2(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3)



void concon3(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3)



**3.Текст программы**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <iomanip>

#include "Windows.h"

using namespace std;

struct list {

int item;

list\* next;

};

void listnum1(list\* li, int& \_count1);

void listnum2(list\* li, int& \_count2);

void listnum3(list\* li, int& \_count3);

void inition(list\* li, int k);

void push(list\*& li, int k, int& count);

void add(list\*& li, int& count);

void pop(list\*& li, int k, int& count);

void deleLI(list\*& li, int& count);

void output(list\* li, int& count);

void searchByIndex(list\* li, int& count);

int searchByValue(list\* li, int value, int& count);

void search(list\*& li, int& count);

void clear(list\*& li, int& count);

void con2(list\*& li1, list\*& li2, int& count1, int& count2);

void concon2(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3);

void con3(list\*& li1, list\*& li2, list\*& li3, int& count1, int& count2, int& count3);

void concon3(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int n = 3;

list\* \_List = new list[n];

bool f = true;

int choice, \_count1 = 1, \_count2 = 1, \_count3 = 1;

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

\_List[i].item = i + 1;

\_List[i].next = NULL;

inition(&\_List[i], (i + 1));

}

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Список № 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Список № 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Список № 3 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Конкотенация из двух списков |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Конкотенация из трех списков |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Выход |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: listnum1(&\_List[0], \_count1); break;

case 2: listnum2(&\_List[1], \_count2); break;

case 3: listnum3(&\_List[2], \_count3); break;

case 4: concon2(&\_List[0], &\_List[1], &\_List[2], \_count1, \_count2, \_count3); break;

case 5: concon3(&\_List[0], &\_List[1], &\_List[2], \_count1, \_count2, \_count3); break;

case 6: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

return 0;

}

void listnum1(list\* li, int& \_count1) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню списка № 1: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Поиск элемента по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Поиск элемента по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Очистить список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(li, \_count1); break;

case 2: output(li, \_count1); break;

case 3: deleLI(li, \_count1); break;

case 4: searchByIndex(li, \_count1); break;

case 5: search(li, \_count1); break;

case 6: clear(li, \_count1); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void listnum2(list\* li, int& \_count2) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню списка № 2: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Поиск элемента по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Поиск элемента по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Очистить список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(li, \_count2); break;

case 2: output(li, \_count2); break;

case 3: deleLI(li, \_count2); break;

case 4: searchByIndex(li, \_count2); break;

case 5: search(li, \_count2); break;

case 6: clear(li, \_count2); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void listnum3(list\* li, int& \_count3) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню списка № 3: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Поиск элемента по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Поиск элемента по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. Очистить список |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(li, \_count3); break;

case 2: output(li, \_count3); break;

case 3: deleLI(li, \_count3); break;

case 4: searchByIndex(li, \_count3); break;

case 5: search(li, \_count3); break;

case 6: clear(li, \_count3); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void inition(list\* li, int k) {

int x;

cout << "Инициализируйте первый элемент для списка № " << k << ":";

cin >> x;

system("cls");

list\* \_li = new list;

\_li->item = x;

li->next = \_li;

li->next->next = NULL;

cout << "Первый элемент со значением = " << x << " инициализирован\n" << endl;

}

void push(list\*& li, int k, int& count) {

list\* \_li = new list;

int j = 1, x;

cout << "Введите новый элемент для добавления:";

cin >> x;

system("cls");

\_li->item = x;

if (k == 1) {

\_li->next = li->next;

li->next = \_li;

}

else if (k == (count + 1)) {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

}

else {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k - 1) {

\_li->next = \_LI->next;

\_LI->next = \_li;

break;

}

j++;

\_LI = \_LI->next;

}

}

count++;

cout << "Элемент " << x << " успешно добавлен";

}

void add(list\*& li, int& count) {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню добавления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Добавление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "|3. Добавление после указанного элемента|" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс куда добавить:";

cin >> x;

system("cls");

if (x <= 0 || x > (count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, x, count);

cout << " в индекс = " << x << endl;

}

}

}

else if (x == 2) {

if (count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

}

else {

while (f1) {

cout << "Введите значение элемента до которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, check, count);

cout << " до элемента " << x << endl;

}

}

}

}

else if (x == 3) {

if (count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

}

else {

while (f1) {

cout << "Введите значение элемента после которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

push(li, (check + 1), count);

cout << " после элемента " << x << endl;

}

}

}

}

else if (x == 4) {

f = false;

}

else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

void output(list\* li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего выводить!!!" << endl;

}

else {

cout << "Cписок № " << li->item << endl;

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

}

}

void pop(list\*& li, int k, int& count) {

list\* \_li = new list;

int x;

if (k == 1) {

\_li = li->next;

li->next = \_li->next;

x = \_li->item;

cout << "Элемент " << \_li->item << " успешно удален" << endl;

delete \_li;

}

else if (k == count) {

list\* \_LI = new list;

\_LI = li;

while (\_LI->next != 0) {

\_li = \_LI;

\_LI = \_LI->next;

}

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

\_li->next = 0;

}

else {

int j = 1;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k) {

\_li->next = \_LI->next;

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

break;

}

j++;

\_li = \_LI;

\_LI = \_LI->next;

}

}

count--;

if (count == 0) {

li->next = 0;

}

}

void deleLI(list\*& li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего удалять!!!" << endl;

}

else {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню удаления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Удаление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Удаление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление после указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Удаление по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс откуда удалить:";

cin >> x;

system("cls");

if (x <= 0 || x > (count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

pop(li, x, count);

}

}

}

else if (x == 2) {

while (f1) {

cout << "Введите значение элемента до которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

if (check == 1) {

cout << "Нельзя удалить до " << x << " \nПотому что там ничего нет!!!" << endl;

}

else {

pop(li, (check - 1), count);

}

}

}

}

else if (x == 3) {

while (f1) {

cout << "Введите значение элемента после которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

if (check == count) {

cout << "Нельзя удалить после " << x << " \nПотому что там ничего нет!!!" << endl;

}

else {

pop(li, (check + 1), count);

}

}

}

}

else if (x == 4) {

while (f1) {

cout << "Введите значение элемента который будем удалять:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

f = false;

pop(li, check, count);

}

}

}

else if (x == 5) {

f = false;

}

else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

}

void searchByIndex(list\* li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

}

else {

int x, i = 1;

bool f = true;

list\* \_li = new list;

while (f) {

cout << "Введите индекс искоемого элемента:";

cin >> x;

system("cls");

if (x <= 0 || x > (count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

}

else {

f = false;

\_li = li->next;

while (\_li != 0) {

if (i == x) {

cout << "Искомый элемент:" << endl;

cout << x << "| " << \_li->item << endl;

}

i++;

\_li = \_li->next;

}

}

}

}

}

int searchByValue(list\* li, int value, int& count) {

int i = 1, j, kol = 0, k = 0, x;

bool f = true;

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

kol++;

j = i;

}

i++;

\_li = \_li->next;

}

if (kol == 0) {

return 0;

}

else if (kol == 1) {

return j;

}

else {

while (f) {

for (i = 0; i < kol; i++) {

cout << i + 1 << "\t|" << value << endl;

}

cout << "Мы нашли " << kol << " элемента с введённым значение, выберите 1 из них:";

cin >> x;

system("cls");

if (x <= 0 || x > kol) {

cout << "Ошибка при вводе!!!" << endl;

}

else {

f = false;

i = 1;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

j = i;

k++;

}

if (k == x) {

break;

}

i++;

\_li = \_li->next;

}

return j;

}

}

}

return 0;

}

void search(list\*& li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего искать!!!" << endl;

}

else {

bool f1 = true;

int search, x;

while (f1) {

cout << "Введите значение элемента который ищем:";

cin >> x;

system("cls");

search = searchByValue(li, x, count);

if (search == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

}

else {

f1 = false;

cout << "Искомый элемент: " << endl << search << "| " << x << endl;

}

}

}

}

void clear(list\*& li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего очищать!!!" << endl;

}

else {

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

list\* \_LI = new list;

\_LI = \_li;

\_li = \_li->next;

delete \_LI;

}

count = 0;

li->next = 0;

cout << "Лист успешно очищен!!!" << endl;

}

}

void con2(list\*& li1, list\*& li2, int& count1, int& count2) {

if (count1 == 0 && count2 == 0) {

cout << "Оба списка не заполнены!!!" << endl;

} else if (count1 == 0) {

cout << li1->item << " не заполнен!!!" << endl;

} else if (count2 == 0) {

cout << li2->item << " не заполнен!!!" << endl;

} else {

cout << li2->item << " список присоединен к " << li1->item << ", а " << li2->item << " пуст!!!" << endl;

list\* \_li = new list;

\_li = li1->next;

while (\_li->next != 0) {

\_li = \_li->next;

}

\_li->next = li2->next;

li2->next = 0;

count1 += count2;

count2 = 0;

}

}

void concon2(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню конкетенации из двух списков: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. 1 + 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. 1 + 3 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. 2 + 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. 2 + 3 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. 3 + 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. 3 + 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: con2(li1, li2, count1, count2); break;

case 2: con2(li1, li3, count1, count3); break;

case 3: con2(li2, li1, count2, count1); break;

case 4: con2(li2, li3, count2, count3); break;

case 5: con2(li3, li1, count3, count1); break;

case 6: con2(li3, li2, count3, count2); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void con3(list\*& li1, list\*& li2, list\*& li3, int& count1, int& count2, int& count3) {

if (count1 == 0 && count2 == 0 && count3 == 0) {

cout << "Все списки не заполнены!!!" << endl;

} else if (count1 == 0) {

cout << li1->item << " не заполнен!!!" << endl;

} else if (count2 == 0) {

cout << li2->item << " не заполнен!!!" << endl;

} else if (count3 == 0) {

cout << li3->item << " не заполнен!!!" << endl;

} else {

cout << li2->item << " и " << li3->item << " список присоединен к " << li1->item << ", а " << li2->item << " и " << li3->item << " пусты!!!" << endl;

list\* \_li = new list;

\_li = li1->next;

while (\_li->next != 0) {

\_li = \_li->next;

}

\_li->next = li2->next;

\_li = li2->next;

while (\_li->next != NULL) {

\_li = \_li->next;

}

\_li->next = li3->next;

count1 = count1 + count2 + count3;

li2->next = 0;

li3->next = 0;

count2 = 0;

count3 = 0;

}

}

void concon3(list\* li1, list\* li2, list\* li3, int& count1, int& count2, int& count3) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню конкетенации из трех списков: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. 1 + 2 + 3 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. 1 + 3 + 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. 2 + 1 + 3 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. 2 + 3 + 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. 3 + 1 + 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6. 3 + 2 + 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 7. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: con3(li1, li2, li3, count1, count2, count3); break;

case 2: con3(li1, li3, li2, count1, count3, count2); break;

case 3: con3(li2, li1, li3, count2, count1, count3); break;

case 4: con3(li2, li3, li1, count2, count3, count3); break;

case 5: con3(li3, li1, li2, count3, count1, count2); break;

case 6: con3(li3, li1, li2, count3, count1, count2); break;

case 7: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

**1.Постановка задачи**

Разработать алгоритм, блок схему и программный код для создания двух списков: первая должна содержать четные числа, а вторая -нечетные числа. Слить эти списки так, чтобы информационные части были упорядочены следующим образом – четное число, нечетное число.

**2. Описание входных и выходных данных**

**void pushe(list\*& li, int value, int& count); -** функция для добавления в список;

**void merger(list\* li1, list\* li2, int& count1, int& count2); -** функция для сортировки;

void pushe(list\*& li, int value, int& count)



void merger(list\* li1, list\* li2, int& count1, int& count2)



**3.Текст программы**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <iomanip>

#include "Windows.h"

using namespace std;

struct list {

int item;

list\* next;

};

void listnum1(list\* li, int& \_count1);

void listnum2(list\* li, int& \_count2);

void inition(list\* li, int k);

void push(list\*& li, int k, int& count);

void add(list\*& li, int& count);

void pop(list\*& li, int k, int& count);

void deleLI(list\*& li, int& count);

void output(list\* li, int& count);

int searchByValue(list\* li, int value, int& count);

void pushe(list\*& li, int value, int& count);

void merger(list\* li1, list\* li2, int& count1, int& count2);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int n = 2;

list\* \_List = new list[n];

bool f = true;

int choice, \_count1 = 1, \_count2 = 1;

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

\_List[i].item = i + 1;

\_List[i].next = NULL;

inition(&\_List[i], (i + 1));

}

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Список № 1 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Список № 2 |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Слияние |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Выход |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: listnum1(&\_List[0], \_count1); break;

case 2: listnum2(&\_List[1], \_count2); break;

case 3: merger(&\_List[0], &\_List[1], \_count1, \_count2); break;

case 4: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

return 0;

}

void listnum1(list\* li, int& \_count1) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню списка № 1: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(li, \_count1); break;

case 2: output(li, \_count1); break;

case 3: deleLI(li, \_count1); break;

case 4: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void listnum2(list\* li, int& \_count2) {

bool f = true;

int choice;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню списка № 2: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: add(li, \_count2); break;

case 2: output(li, \_count2); break;

case 3: deleLI(li, \_count2); break;

case 4: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

}

void inition(list\* li, int k) {

int x;

cout << "Инициализируйте первый элемент для списка № " << k << ":";

cin >> x;

system("cls");

list\* \_li = new list;

\_li->item = x;

li->next = \_li;

li->next->next = NULL;

cout << "Первый элемент со значением = " << x << " инициализирован\n" << endl;

}

void push(list\*& li, int k, int& count) {

list\* \_li = new list;

int j = 1, x;

cout << "Введите новый элемент для добавления:";

cin >> x;

system("cls");

\_li->item = x;

if (k == 1) {

\_li->next = li->next;

li->next = \_li;

} else if (k == (count + 1)) {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

} else {

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k - 1) {

\_li->next = \_LI->next;

\_LI->next = \_li;

break;

}

j++;

\_LI = \_LI->next;

}

}

count++;

cout << "Элемент " << x << " успешно добавлен";

}

void add(list\*& li, int& count) {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню добавления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Добавление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "|3. Добавление после указанного элемента|" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс куда добавить:";

cin >> x;

system("cls");

if (x <= 0 || x > (count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, x, count);

cout << " в индекс = " << x << endl;

}

}

} else if (x == 2) {

if (count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

} else {

while (f1) {

cout << "Введите значение элемента до которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, check, count);

cout << " до элемента " << x << endl;

}

}

}

} else if (x == 3) {

if (count <= 0) {

cout << "Лист пуст, нельзя добавить до или после!!!" << endl;

} else {

while (f1) {

cout << "Введите значение элемента после которого добавить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

push(li, (check + 1), count);

cout << " после элемента " << x << endl;

}

}

}

} else if (x == 4) {

f = false;

} else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

void output(list\* li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего выводить!!!" << endl;

} else {

cout << "Cписок № " << li->item << endl;

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

}

}

void pop(list\*& li, int k, int& count) {

list\* \_li = new list;

int x;

if (k == 1) {

\_li = li->next;

li->next = \_li->next;

x = \_li->item;

cout << "Элемент " << \_li->item << " успешно удален" << endl;

delete \_li;

} else if (k == count) {

list\* \_LI = new list;

\_LI = li;

while (\_LI->next != 0) {

\_li = \_LI;

\_LI = \_LI->next;

}

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

\_li->next = 0;

} else {

int j = 1;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI != 0) {

if (j == k) {

\_li->next = \_LI->next;

x = \_LI->item;

cout << "Элемент " << \_LI->item << " успешно удален" << endl;

delete \_LI;

break;

}

j++;

\_li = \_LI;

\_LI = \_LI->next;

}

}

count--;

if (count == 0) {

li->next = 0;

}

}

void deleLI(list\*& li, int& count) {

if (count <= 0) {

cout << "Лист пуст, нечего удалять!!!" << endl;

} else {

bool f = true, f1 = true;

int x, check;

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Меню удаления: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Удаление по индексу |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Удаление до указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Удаление после указанного элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Удаление по значению |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Назад |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

if (x == 1) {

while (f1) {

cout << "Введите индекс откуда удалить:";

cin >> x;

system("cls");

if (x <= 0 || x > (count + 1)) {

cout << "Вы вышли за границы списка!!!" << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

pop(li, x, count);

}

}

} else if (x == 2) {

while (f1) {

cout << "Введите значение элемента до которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

if (check == 1) {

cout << "Нельзя удалить до " << x << " \nПотому что там ничего нет!!!" << endl;

} else {

pop(li, (check - 1), count);

}

}

}

} else if (x == 3) {

while (f1) {

cout << "Введите значение элемента после которого удалить:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

if (check == count) {

cout << "Нельзя удалить после " << x << " \nПотому что там ничего нет!!!" << endl;

} else {

pop(li, (check + 1), count);

}

}

}

} else if (x == 4) {

while (f1) {

cout << "Введите значение элемента который будем удалять:";

cin >> x;

system("cls");

check = searchByValue(li, x, count);

if (check == 0) {

cout << "Введенного элемента не существует." << endl << "Повторите заново ввод!!!" << endl;

} else {

f1 = false;

f = false;

pop(li, check, count);

}

}

} else if (x == 5) {

f = false;

} else {

cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl;

}

}

}

}

int searchByValue(list\* li, int value, int& count) {

int i = 1, j, kol = 0, k = 0, x;

bool f = true;

list\* \_li = new list;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

kol++;

j = i;

}

i++;

\_li = \_li->next;

}

if (kol == 0) {

return 0;

} else if (kol == 1) {

return j;

} else {

while (f) {

for (i = 0; i < kol; i++) {

cout << i + 1 << "\t|" << value << endl;

}

cout << "Мы нашли " << kol << " элемента с введённым значение, выберите 1 из них:";

cin >> x;

system("cls");

if (x <= 0 || x > kol) {

cout << "Ошибка при вводе!!!" << endl;

} else {

f = false;

i = 1;

\_li = li->next;

while (\_li != 0) {

if (\_li->item == value) {

j = i;

k++;

}

if (k == x) {

break;

}

i++;

\_li = \_li->next;

}

return j;

}

}

}

return 0;

}

void pushe(list\*& li, int value, int& count) {

list\* \_li = new list;

\_li->item = value;

int j = 1;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

count++;

}

void merger(list\* li1, list\* li2, int& count1, int& count2) {

list\* \_li1 = new list; //четные 2 4 6 8

list\* \_li2 = new list; //не четные 1 3 5 7

\_li1 = li1->next; // 2 1 4 3 6 5 8 7

\_li2 = li2->next;

int i = 0, n = count1 + count2, chet = 0, nechet = 1;

int arr[100] = { 0 }, arr2[100] = { 0 };

while (\_li1 != 0) {

arr[i] = \_li1->item;

\_li1 = \_li1->next;

i++;

}

while (\_li2 != 0) {

arr[i] = \_li2->item;

\_li2 = \_li2->next;

i++;

}

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

if ((arr[i] % 2) == 0) {

arr2[chet] = arr[i];

chet += 2;

} else {

arr2[nechet] = arr[i];

nechet += 2;

}

}

li1->next = 0;

li2->next = 0;

count1 = 0;

count2 = 0;

list\* \_li = new list;

\_li->item = arr[0];

li1->next = \_li;

li1->next->next = NULL;

count1++;

for (i = 1; i < n; i++) {

pushe(li1, arr[i], count1);

}

}

**1.Постановка задачи**

Разработать программу для задачи - сформировать однонаправленный список из вводимой строки. В поле каждого элемента списка записать отдельный символ. Если первый символ - буква "А", то в конец списка добавить еще одну букву "А", иначе из списка исключить все буквы "А". Вывести полученный результат.

**2. Описание входных и выходных данных**

**void push(list\*& li); -** функция инициализации/добавление;

**void output(list\* li); -** функция вывода на экран списка;

**void check(list\*& li); -** функция которая проверяет и удаляет “A” или “a”;

**int search(list\* li); -** функция которая ищет кол – во “A” или “a”;

**int search\_A\_a(list\* li); -** функция находящая “A” или “a”;

int main()



void push(list\*& li)



void output(list\* li)



int search\_A\_a(list\* li)



int search(list\* li)



void check(list\*& li)



**3.Текст программы**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <iomanip>

#include "Windows.h"

using namespace std;

struct list {

string item;

list\* next;

};

int \_count = 0, \_check;

bool ifcheck = false;

void push(list\*& li);

void output(list\* li);

void check(list\*& li);

int search(list\* li);

int search\_A\_a(list\* li);

int main() {

setlocale(0, "rus");

char s[54];

list\* \_List = new list;

bool f = true;

int choice, k = 0;

\_List->item = "#4";

\_List->next = 0;

push(\_List);

while (f) {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Добавление элемента |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Вывод элементов |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Проверка списка |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Выход |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер операции:";

cin >> choice;

system("cls");

switch (choice) {

case 1: push(\_List); break;

case 2: output(\_List); break;

case 3: check(\_List); break;

case 4: f = false; break;

default: cout << "Такой операции не существует!!!" << endl << "Введите заново." << endl; break;

}

}

return 0;

}

void push(list\*& li) {

string x;

if (\_count == 0) {

cout << "Инициализируйте первый элемент для списка:";

cin >> x;

if (x == "A") {

ifcheck = true;

\_check = 1;

}

else if (x == "a") {

ifcheck = true;

\_check = 2;

}

system("cls");

list\* \_li = new list;

\_li->item = x;

li->next = \_li;

li->next->next = NULL;

cout << "Первый элемент со значением = " << x << " инициализирован\n" << endl;

}

else {

cout << "Введите новый элемент для добавления:";

cin >> x;

list\* \_li = new list;

\_li->item = x;

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

cout << "Элемент = " << x << " добавлен\n" << endl;

}

\_count++;

}

void output(list\* li) {

if (\_count <= 0) {

cout << "Лист пуст, нечего выводить!!!" << endl;

}

else {

int i = 1;

list\* \_li = li->next;

cout << "------------------------------------" << endl;

cout << "|\tЭлементы\t: Индекс |\n";

cout << "------------------------------------" << endl;

while (\_li != 0) {

cout << "|\t" << \_li->item << "\t\t: " << i << "\t |" << endl;

cout << "------------------------------------" << endl;

\_li = \_li->next;

i++;

}

}

}

int search\_A\_a(list\* li) {

int kol = 0, i = 1, j = 0;

list\* \_li = new list;

\_li = li->next;

while (\_li != NULL) {

if (\_li->item == "A" || \_li->item == "a") {

kol++;

j = i;

}

i++;

\_li = \_li->next;

}

if (kol == 0) {

return 0;

}

else if (kol == 1) {

return j;

}

else {

while (\_li != 0) {

if (\_li->item == "A" || \_li->item == "a") {

j = i;

}

\_li = \_li->next;

}

return j;

}

return 0;

}

int search(list\* li) {

list\* \_li = new list;

int k = 0;

\_li = li->next;

while (\_li != NULL) {

if (\_li->item == "A" || \_li->item == "a") {

k++;

}

\_li = \_li->next;

}

return k;

}

void check(list\*& li) {

if (ifcheck == true) {

list\* \_li = new list;

if (\_check == 1) {

\_li->item = "A";

}

else if (\_check == 2) {

\_li->item = "a";

}

list\* \_LI = new list;

\_LI = li->next;

while (\_LI->next != 0) {

\_LI = \_LI->next;

}

\_LI->next = \_li;

\_li->next = 0;

}

else {

int c, k;

string trash;

k = search(li);

for (int l = 0; l < k; l++) {

c = search\_A\_a(li);

if (c == 0) {

cout << "В списке нету букв <<A>> или <<а>>!!!" << endl;

}

else if (c == \_count) {

list\* \_li = new list;

list\* \_LI = new list;

\_li = li;

while (\_li->next != 0) {

\_LI = \_li;

\_li = \_li->next;

}

trash = \_li->item;

delete \_li;

\_LI->next = 0;

}

else {

list\* \_li = new list;

list\* \_LI = new list;

int j = 1;

\_LI = li->next;

while (\_LI != 0) {

if (j == c) {

\_li->next = \_LI->next;

trash = \_LI->item;

delete \_LI;

break;

}

j++;

\_li = \_LI;

\_LI = \_LI->next;

}

}

\_count--;

}

}

}