Организация списков с помощью указателей и структур.

Дан список идентификаторов. Длина каждого иден тификатора не более 8 символов. Идентификаторы в списке расположены в лексикографическом порядке. Составить функции (подпрограммы) для следующих операций:

1.Удалить из списка заданный идентификатор.

2.Заменить на заданный идентификатор значение k-го по порядку элемента списка.

3.Определить количество идентификаторов в списке начинающихся с заданной буквы.

Код:

#include <iostream>

#include<string>

#include <windows.h>

using namespace std;

struct EL\_SP

{

string id;

EL\_SP \*sled;

};

void Vkl(EL\_SP \*\*p, string t\_id)

{

EL\_SP \*pt;

EL\_SP \*k = NULL;

EL\_SP \*j = NULL;

pt = new EL\_SP();

pt->id = t\_id;

if (\*p == NULL || pt->id<(\*p)->id)

{

pt->sled = \*p;

\*p = pt;

}

else

{

k = \*p;

while (k != NULL&&pt->id >= k->id)

{

j = k;

k = k->sled;

}

j->sled = pt;

pt->sled = k;

}

}

void PechSp(EL\_SP \*p)

{

EL\_SP \*i;

cout << "Результат: " << endl;

for (i = p; i != NULL; i = i->sled)

cout << i->id << endl;

}

void zamenak(EL\_SP \*p, int kol1)

{

EL\_SP \*i;

int k;

cout << "ВВедите какой элемент по номеру заменить";

cin >> k;

cin.ignore();

string zd\_id;

cout << "ВВедите на какой элемент заменить";

cin >> zd\_id;

int j = 1;

if (kol1 < k)

cout << "вы превысили длину списка";

for (i = p,j; i != NULL&&j<k; i = i->sled,j++)

continue;

i->id=zd\_id;

cout<<"замена выполнена"<<endl;

}

void udaleniefivezadan(EL\_SP \*\*p, int \*kol1)

{

string zd\_id;

int kol=\*kol1;

int j = 1;

cout << "ВВедите идентификатор который нужно удалить";

getline(cin, zd\_id);

EL\_SP \*i = \*p, \*j1=NULL;

while (i != NULL)

{

if (i->id == zd\_id)

{

if (j == 1)

{

\*p = (\*p)->sled;

delete i;

\*kol1 -= 1;

cout << "удаление выполнено " << endl;

break;

}

else

{

j1->sled = i->sled;

delete i;

\*kol1 -= 1;

cout << "удаление выполнено " << endl;

break;

}

}

else

{

j1 = i;

i = i->sled;

j++;

}

}

if (\*kol1 == kol)

cout << "Подходящий элемент не найден!" << endl;

}

void kolnachnachtoto(EL\_SP \*p )

{

string zd\_id;

cout << "ВВедите первую букву идентификатора";

cin>>zd\_id;

int kolvo=0;

EL\_SP \*i;

i=p;

while (i != NULL)

{

if (i->id[0] == zd\_id[0])

{

kolvo++;

}

i = i->sled;

}

cout << "Количество идентификаторов на " <<zd\_id<<" = "<<kolvo<< endl;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int kol;

char n;

EL\_SP \*p;

unsigned i;

string t\_id;

p = NULL;

cout << "Введите индентифактор" << endl << "После каждого нажмите ENTER" << endl;

for (i = 1;; i++)

{

getline(cin, t\_id);

cin.sync();

cin.clear();

if (t\_id.size() > 8)

{

i--;

cout << "Ошибка!" << endl << "Введите индентификатор не более 8 символов" << endl;

}

else if (t\_id.size() == 0)

{

i--;

cout << "Ввод индентификаторов закончен" << endl;

break;

}

else

Vkl(&p, t\_id);

}

kol = i;

cout << kol;

do

{

cout << "========================================================" << endl;

cout << "Выберите номер пункта меню:" << endl;

cout << "1 - вывод идентификаторов в алфавитном порядке" << endl;

cout << "2 - удалить заданный идентификатор" << endl;

cout << "3 - заменить на заданный идентификатор k-ого элемента" << endl;

cout << "4 - кол-во идентификаторов на заданный символ" << endl;

cout << "5 - выход" << endl;

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

n = cin.get();

cin.sync();

switch (n)

{

case '1': PechSp(p); break;

case '2': udaleniefivezadan(&p,&kol); break;

case '3': zamenak(p,kol); break;

case '4': kolnachnachtoto(p); break;

default: cout << "\nНужно вводить номер пункта от 1 до 5";

}

if (n != '5')

{

cout << "\nДля продолжения нажмите любую клавишу";

cin.get();

}

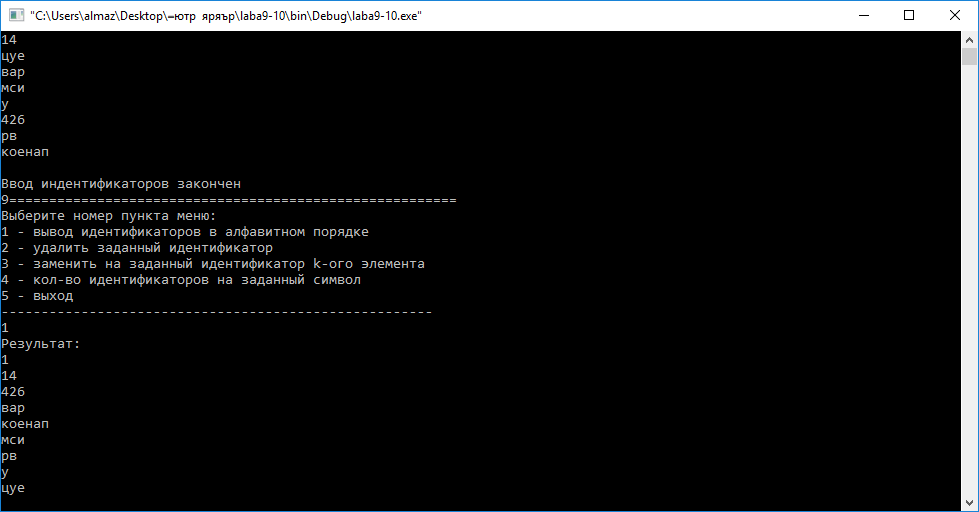
} while (n != '5');

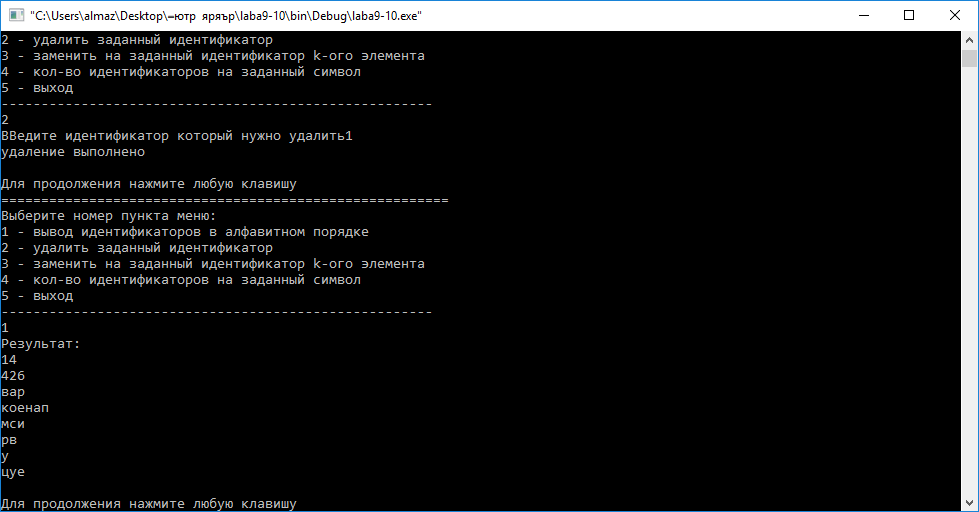
cout << endl;

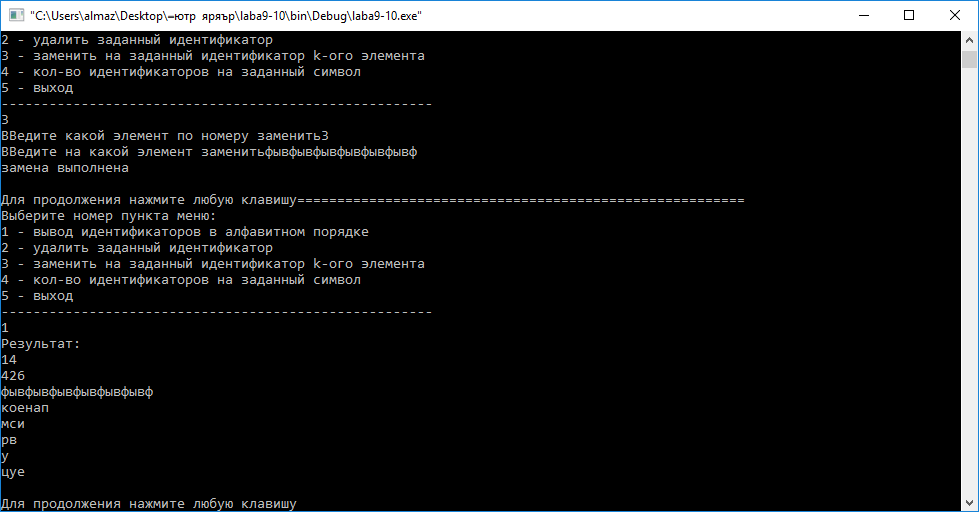
return 1;

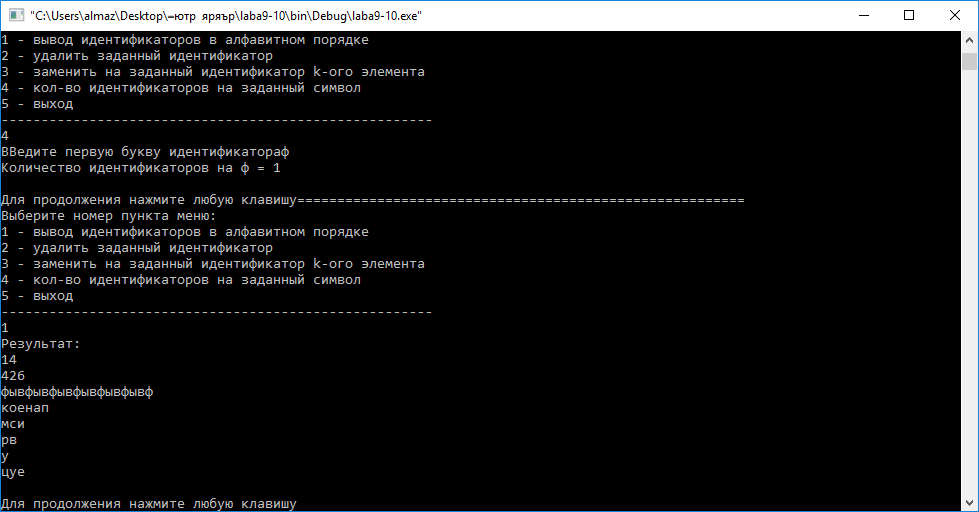
}

Результаты:









Модульное программирование.

Записать в отдельные файлы свои функции обработки списка. Добавить в проект имена своих файлов. Изменить функцию

main()

в файле «main.c», чтобы вызывались ваши функции и выводились

результаты их работы. Проверить результат работы программы.

Изменить в файле проекта расширения своих файлов на .obj

и снова выполнить программу.

#include "head.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int kol;

char n;

EL\_SP \*p;

unsigned i;

string t\_id;

p = NULL;

cout << "Введите индентифактор" << endl << "После каждого нажмите ENTER" << endl;

for (i = 1;; i++)

{

getline(cin, t\_id);

cin.sync();

cin.clear();

if (t\_id.size() > 8)

{

i--;

cout << "Ошибка!" << endl << "Введите индентификатор не более 8 символов" << endl;

}

else if (t\_id.size() == 0)

{

i--;

cout << "Ввод индентификаторов закончен" << endl;

break;

}

else

Vkl(&p, t\_id);

}

kol = i;

cout << kol;

do

{

cout << "========================================================" << endl;

cout << "Выберите номер пункта меню:" << endl;

cout << "1 - вывод идентификаторов в алфавитном порядке" << endl;

cout << "2 - удалить заданный идентификатор" << endl;

cout << "3 - заменить на заданный идентификатор k-ого элемента" << endl;

cout << "4 - кол-во идентификаторов на заданный символ" << endl;

cout << "5 - выход" << endl;

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

n = cin.get();

cin.sync();

switch (n)

{

case '1': PechSp(p); break;

case '2': udaleniefivezadan(&p,&kol); break;

case '3': zamenak(p,kol); break;

case '4': kolnachnachtoto(p); break;

case '5': break;

default: cout << "\nНужно вводить номер пункта от 1 до 5";

}

if (n != '5')

{

cout << "\nДля продолжения нажмите любую клавишу";

cin.get();

}

} while (n != '5');

cout << endl;

return 1;

}

#pragma once

#include<iostream>

#include<string>

#include<windows.h>

using namespace std;

struct EL\_SP

{

string id;

EL\_SP \*sled;

};

void Vkl(EL\_SP \*\*p, string t\_id);

void PechSp(EL\_SP \*p);

void udaleniefivezadan(EL\_SP \*\*p, int \*kol1);

void zamenak(EL\_SP \*p, int kol1);

void kolnachnachtoto(EL\_SP \*p );

#include "head.h"

void Vkl(EL\_SP \*\*p, string t\_id)

{

EL\_SP \*pt;

EL\_SP \*k = NULL;

EL\_SP \*j = NULL;

pt = new EL\_SP();

pt->id = t\_id;

if (\*p == NULL || pt->id<(\*p)->id)

{

pt->sled = \*p;

\*p = pt;

}

else

{

k = \*p;

while (k != NULL&&pt->id >= k->id)

{

j = k;

k = k->sled;

}

j->sled = pt;

pt->sled = k;

}

}

#include "head.h"

void kolnachnachtoto(EL\_SP \*p )

{

string zd\_id;

cout << "ВВедите первую букву идентификатора";

cin>>zd\_id;

int kolvo=0;

EL\_SP \*i;

i=p;

while (i != NULL)

{

if (i->id[0] == zd\_id[0])

{

kolvo++;

}

i = i->sled;

}

cout << "Количество идентификаторов на " <<zd\_id<<" = "<<kolvo<< endl;

}

#include "head.h"

void zamenak(EL\_SP \*p, int kol1)

{

EL\_SP \*i;

int k;

cout << "ВВедите какой элемент по номеру заменить";

cin >> k;

cin.ignore();

string zd\_id;

cout << "ВВедите на какой элемент заменить";

cin >> zd\_id;

int j = 1;

if (kol1 < k)

cout << "вы превысили длину списка";

for (i = p,j; i != NULL&&j<k; i = i->sled,j++)

continue;

i->id=zd\_id;

cout<<"замена выполнена"<<endl;

}

#include "head.h"

void udaleniefivezadan(EL\_SP \*\*p, int \*kol1)

{

string zd\_id;

int kol=\*kol1;

int j = 1;

cout << "ВВедите идентификатор который нужно удалить";

getline(cin, zd\_id);

EL\_SP \*i = \*p, \*j1=NULL;

while (i != NULL)

{

if (i->id == zd\_id)

{

if (j == 1)

{

\*p = (\*p)->sled;

delete i;

\*kol1 -= 1;

cout << "удаление выполнено " << endl;

break;

}

else

{

j1->sled = i->sled;

delete i;

\*kol1 -= 1;

cout << "удаление выполнено " << endl;

break;

}

}

else

{

j1 = i;

i = i->sled;

j++;

}

}

if (\*kol1 == kol)

cout << "Подходящий элемент не найден!" << endl;

}

#include "head.h"

void PechSp(EL\_SP \*p)

{

EL\_SP \*i;

cout << "Результат: " << endl;

for (i = p; i != NULL; i = i->sled)

cout << i->id << endl;

}

