Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №14**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Методы поиска: метод Кнутта-Мориса Пратта, метод Бойера-Мура

Вариант 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2021

**Цель работы**

Получить практические навыки работы с методом поиска Кнута-Мориса-Пратта и методом поиска Бойера-Мура.

**Постановка задачи**

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 100 элементов. Для заполнения элементов массива использовать ДСЧ.
2. Предусмотреть сохранение массива в файл и загрузку массива из файла.
3. Предусмотреть возможность добавления и удаления элементов из массива (файла).
4. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать метод Бойера-Мура и метод Кнута-Морриса-Пратта.

**Анализ задачи**

1. Для решения задачи необходимо:

1.1. Создать функцию, формирующую массив и заполняющую массив:

struct Massiv{

string fio;

int d,m,g;

string adres;

};

Massiv \*m;

void Create(){}

1.2. Создать функцию, производящую поиск элемента методом Кнута-Морриса-Пратта:

void KMP(){

string kucha, isk;

for (int i=0;i<r;i++) kucha+=m[i].adres+'\n';

cout << kucha << endl; //!\_!

cout << "Адрес искомого элемента: "; cin.ignore(30,'\n'); getline(cin,isk);

int \*s = new int [isk.length()];

for (int i=1,j=0;i<isk.length();)

{

if (isk[i]==isk[j]) {s[i]=j+1;i++;j++;}

else {

if (j==0) s[i++]=0;

else j=s[j-1];

}

}

int i=0,j=0,nom=0;

bool nai=0;

while (i<kucha.length()){

nom=0;

if (kucha[i]==isk[j]){

i++;j++;

if (j==isk.length()){

for (int e=0;e<i;e++) if (kucha[e]=='\n') nom++;

cout << "Элемент, содержащий введённые данные: " << nom+1 << ". " << m[nom].fio << "; "; if (m[nom].d<10) cout << "0"; cout << m[nom].d << "."; if (m[nom].m<10) cout << "0"; cout << m[nom].m << "." << m[nom].g << "; " << m[nom].adres << ";" << endl;

nai=1;

}

}

else{

if (j==0) i++;

else j=s[j-1];

}

}

if (!nai) cout << "Элемент не найден. " << endl;

}

1.3. Создать функцию, производящую поиск элемента в массиве методом Боера-Мура:

void BouerMur(){

string kucha,isk;

int sdv,rms=1,poz,nom=0;

bool end=0,nai=0,sov=0;

for (int i=0;i<r;i++) kucha+=m[i].adres+'\n';

cout << "Адрес искомого элемента: " << endl;

cin.ignore(30,'\n');

getline (cin, isk);

MS \*first=new MS;

MS \*cur = first;

MS \*nov;

cur->next=NULL;

first->sim = isk[isk.length()-1];

first->zns = isk.length();

for (int i=isk.length()-2;i>=-1;i--){

cur = first;

nai=0;

while (cur->next!=NULL&&!nai){

if (isk[i]==cur->sim) nai=1;

cur=cur->next;

}

if (!nai){

nov = new MS;

nov->next=NULL;

nov->sim=isk[i];

nov->zns=rms++;

cur->next=nov;

cur=nov;

}

}

cur = first;

while (cur->next!=NULL) {

cout << "Символ: " << cur->sim << " Сдвиг: " << cur->zns << endl;

cur=cur->next;

}

poz=isk.length()-1;

nai=0;

cur=first;

int elm,kel;

bool tak=0;

while(poz<kucha.length()){

elm=isk.length()-1;

cur=first;

tak=0;

nom=0;

if (kucha[poz]==isk[elm]){

kel=poz;

while (kucha[kel]==isk[elm]&&elm>=0){

kel--;elm--;

}

if (elm==-1){sov=1;nai=1;}

}

for (int j=0;j<poz;j++) if (kucha[j]=='\n') nom++;

while (cur->next!=NULL&&!tak) {

if (kucha[poz]==cur->sim) tak=1;

else cur=cur->next;

}

if (tak){

if (poz+cur->zns<kucha.length()) poz+=cur->zns;

else poz=kucha.length();

}

else{

if (poz+isk.length()<kucha.length()) poz+=isk.length();

else poz=kucha.length();

}

if (sov){

cout << "Элемент, содержащий введённые данные: " << nom+1 << ". " << m[nom].fio << "; "; if (m[nom].d<10) cout << "0"; cout << m[nom].d << "."; if (m[nom].m<10) cout << "0"; cout << m[nom].m << "." << m[nom].g << "; " << m[nom].adres << ";" << endl;

}

sov=0;

}

if (!nai) cout << "Элемент не найден. " << endl;

}

1.4. Сформировать меню, вызывающее выбранную функцию:

while (v!=0){

cout << "Выберете действие: " << endl << "0. Завершить работу программы. \n1. Сформировать массив случайно; \n2. Сформировать массив самостоятельно; \n3. Скопировать массив из файла; \n4. Удалить элемент из массива; \n5. Добавить элемент в массив; ";

cout <<"\n6. Запустить метод поиска Боера-Мура; \n7. Запустить метод поиска Кнута-Морриса-Пратта. " << endl;

cin >> v;

switch(v){

case 0: return 0;

case 1: Create();break;

case 2: CUser(); break;

case 3: Take(); break;

case 4: if (mc==true) Udalit(); else cout << "База данных отсутствует. " << endl; break;

case 5: if (mc==true) Dop(); else cout << "База данных отсутствует. " << endl; break;

case 6: if (mc==true) BouerMur(); else cout << "База данных отсутствует. " << endl; break;

case 7: if (mc==true) KMP(); else cout << "База данных отсутствует. " << endl; break;

default: cout << "Неверный ввод! " << endl;

}

cout << endl;

}

2. В ходе работы были использованы следующие типы данных:

2.1. Для получения данных из файла используется класс ifstream:  
ifstream ibaza("baza.txt");

2.2. Для вывода данных в файл используется класс ofstream:  
ofstream obaza("baza.txt");

2.3. Для хранения числа элементов и поиска элемента по номеру используется тип int:

int r, i;

3. Данные для решения задачи представлены в следующем виде:

3.1. Для хранения данных элементов использовалась структура данных:

struct Massiv{

string fio;

int d,m,g;

string adres;

};

3.2. Поля структуры имеют строковый тип string для хранения ФИО и адреса элемента и целочисленный тип int для хранения даты рождения элемента.

4. Для операций ввода и вывода использовались следующие операторы:

4.1. Для ввода данных использовался оператор cin и функция getline():

cin >> r; getline (cin, isk);

4.2. Для вывода значений на экран используется оператор cout:

cout << "Адрес искомого элемента: " << endl;

5. Для решения задач использовались следующие циклы:

5.1. Цикл while для проверки правильности ввода:

cin >> nm[nomer-1].g;

while (nm[nomer-1].g<1900||nm[nomer-1].g>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

cin >> nm[nomer-1].g;

}

5.2. Для вывода данных в файл используется цикл for:

for (int i=0;i<nomer-1;i++){

nm[i]=m[i];

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

**Полный код программы**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <iomanip>

using namespace std;

struct Massiv{

string fio;

int d,m,g;

string adres;

};

Massiv \*m;

int n,ran,v=-1,r;

bool mc=false,nmc=false;

void Create(){

ofstream obaza("baza.txt");

srand(time(0));

cout << "Количество элементов: ";

cin >> r;

m = new Massiv[r];

obaza << "Количество элементов: " << r << endl;

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

n=rand()%10;

ran=rand()%10;

m[i].d=1+rand()%28;

m[i].m=1+rand()%12;

m[i].g=1900+rand()%121;

switch (n){

case 0: m[i].fio="Анастасия "; break;

case 1: m[i].fio="Мария "; break;

case 2: m[i].fio="Дарья "; break;

case 3: m[i].fio="Анна "; break;

case 4: m[i].fio="Елизавета "; break;

case 5: m[i].fio="Александр "; break;

case 6: m[i].fio="Максим "; break;

case 7: m[i].fio="Иван "; break;

case 8: m[i].fio="Артём "; break;

case 9: m[i].fio="Дмиртий "; break;

}

switch (ran){

case 0: if (n<5) m[i].fio+="Владимировна "; else m[i].fio+="Владимирович "; break;

case 1: if (n<5) m[i].fio+="Ивановна "; else m[i].fio+="Иванович "; break;

case 2: if (n<5) m[i].fio+="Александровна "; else m[i].fio+="Александрович "; break;

case 3: if (n<5) m[i].fio+="Сергеевна "; else m[i].fio+="Сергеевич "; break;

case 4: if (n<5) m[i].fio+="Антоновна "; else m[i].fio+="Антонович "; break;

case 5: if (n<5) m[i].fio+="Петровна "; else m[i].fio+="Петрович "; break;

case 6: if (n<5) m[i].fio+="Дмитриевна "; else m[i].fio+="Дмитриевич "; break;

case 7: if (n<5) m[i].fio+="Анатольевна "; else m[i].fio+="Анатольевич "; break;

case 8: if (n<5) m[i].fio+="Артёмовна "; else m[i].fio+="Артёмович "; break;

case 9: if (n<5) m[i].fio+="Семёновна "; else m[i].fio+="Семёнович "; break;

}

ran=rand()%10;

switch (ran){

case 0: if (n<5) m[i].fio+="Иванова"; else m[i].fio+="Иванов"; break;

case 1: if (n<5) m[i].fio+="Петрова"; else m[i].fio+="Петров"; break;

case 2: if (n<5) m[i].fio+="Сидорова"; else m[i].fio+="Сидоров"; break;

case 3: if (n<5) m[i].fio+="Кузнецова"; else m[i].fio+="Кузнецов"; break;

case 4: if (n<5) m[i].fio+="Попова"; else m[i].fio+="Попов"; break;

case 5: if (n<5) m[i].fio+="Смирнова"; else m[i].fio+="Смирнов"; break;

case 6: if (n<5) m[i].fio+="Васильева"; else m[i].fio+="Васильев"; break;

case 7: if (n<5) m[i].fio+="Соколова"; else m[i].fio+="Соколов"; break;

case 8: if (n<5) m[i].fio+="Новикова"; else m[i].fio+="Новиков"; break;

case 9: if (n<5) m[i].fio+="Волкова"; else m[i].fio+="Волков"; break;

}

ran=rand()%10;

switch (ran){

case 0: m[i].adres="Пермь, "; break;

case 1: m[i].adres="Москва, "; break;

case 2: m[i].adres="Санкт-Петербург, "; break;

case 3: m[i].adres="Екатеринбург, "; break;

case 4: m[i].adres="Новосибирск, "; break;

case 5: m[i].adres="Новокузнецк, "; break;

case 6: m[i].adres="Новороссийск, "; break;

case 7: m[i].adres="Омск, "; break;

case 8: m[i].adres="Архангельск, "; break;

case 9: m[i].adres="Курган, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="ул. Екатерининская, "; break;

case 1: m[i].adres+="ул. Мостовая, "; break;

case 2: m[i].adres+="ул. Центральная, "; break;

case 3: m[i].adres+="ул. Королёва, "; break;

case 4: m[i].adres+="ул. Мира, "; break;

case 5: m[i].adres+="ул. Луговая, "; break;

case 6: m[i].adres+="ул. Лаврова, "; break;

case 7: m[i].adres+="ул. Институтская, "; break;

case 8: m[i].adres+="ул. Арбат, "; break;

case 9: m[i].adres+="ул. Революции, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="д. 17"; break;

case 1: m[i].adres+="д. 14"; break;

case 2: m[i].adres+="д. 46"; break;

case 3: m[i].adres+="д. 8"; break;

case 4: m[i].adres+="д. 5-А"; break;

case 5: m[i].adres+="д. 91"; break;

case 6: m[i].adres+="д. 194"; break;

case 7: m[i].adres+="д. 18"; break;

case 8: m[i].adres+="д. 7"; break;

case 9: m[i].adres+="д. 81"; break;

}

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

mc=true;

}

void CUser(){

ofstream obaza("baza.txt");

cout << "Количество элементов: ";

cin >> r;

m = new Massiv[r];

obaza << "Количество элементов: " << r << endl;

cin.ignore(30,'\n');

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

cout << i+1 << "." << endl << "ФИО: ";

getline (cin, m[i].fio);

cout << "День (число месяца) рождения: "; cin >> m[i].d;

cout << "Месяц рождения: " ; cin >> m[i].m;

cout << "Год рождения: "; cin >> m[i].g;

cout << "Адрес: "; cin.ignore(30,'\n'); getline (cin, m[i].adres);

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

}

void Take(){

int i=0;

ifstream ibaza("baza.txt");

if (!ibaza) cout << "Файл не открыт. "; else cout << "Файл открыт успешно. "; cout << endl;

char sod[50],drop[30];

ibaza.getline(drop,30,':');

if (ibaza.eof()) cout << "Файл пустой. " << endl;

else{

ibaza >> r;

m = new Massiv [r];

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

ibaza.getline(drop,5,' ');

ibaza.getline(sod,50,';');

m[i].fio = sod;

ibaza >> m[i].d;

ibaza.getline(drop,5,'.');

ibaza >> m[i].m;

ibaza.getline(drop,5,'.');

ibaza >> m[i].g;

ibaza.getline(drop,30,' ');

ibaza.getline(sod,50,';');

ibaza.getline(drop,30,'\n');

m[i].adres=sod;

cout << i+1 << ". " << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

ibaza.close();

cout << "Файл успешно закрыт. " << endl;

mc=true;

}

}

void Dop(){

ofstream obaza("baza.txt");

int nomer;

cout << "Номер, на позицию которого хотите добавить элемент: ";

cin >> nomer;

while (nomer<1||nomer>r+1||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать длину массива более чем на 1! Повторите ввод: ";

cin >> nomer;

}

r++;

obaza << "Количество элементов: " << r << endl;

Massiv \*nm = new Massiv [r];

for (int i=0;i<nomer-1;i++){

nm[i]=m[i];

cout << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) cout << "0"; cout << nm[i].d << "."; if (nm[i].m<10) cout << "0"; cout << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "ФИО элемента: "; cin.ignore(30,'\n'); getline (cin, nm[nomer-1].fio);

cout << "День (число месяца), в который родился элемент: "; cin >> nm[nomer-1].d;

while (nm[nomer-1].d<1||nm[nomer-1].d>31||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 31! Повторите ввод: ";

cin >> nm[nomer-1].d;

}

cout << "Месяц рождения элемента: ";

cin >> nm[nomer-1].m;

while (nm[nomer-1].m<1||nm[nomer-1].m>12||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 12! Повторите ввод: ";

cin >> nm[nomer-1].m;

}

cout << "Год рождения элемента: ";

cin >> nm[nomer-1].g;

while (nm[nomer-1].g<1900||nm[nomer-1].g>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

cin >> nm[nomer-1].g;

}

cout << "Адрес элемента: "; cin.ignore(30,'\n'); getline (cin, nm[nomer-1].adres);

cout << nomer << ". " << nm[nomer-1].fio << "; "; if (nm[nomer-1].d<10) cout << "0"; cout << nm[nomer-1].d << "."; if (nm[nomer-1].m<10) cout << "0"; cout << nm[nomer-1].m << "." << nm[nomer-1].g << "; " << nm[nomer-1].adres << ";" << endl;

obaza << nomer << ". " << nm[nomer-1].fio << "; "; if (nm[nomer-1].d<10) obaza << "0"; obaza << nm[nomer-1].d << "."; if (nm[nomer-1].m<10) obaza << "0"; obaza << nm[nomer-1].m << "." << nm[nomer-1].g << "; " << nm[nomer-1].adres << ";" << endl;

for (int i=nomer;i<r;i++){

nm[i]=m[i-1];

cout << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) cout << "0"; cout << nm[i].d << "."; if (nm[i].m<10) cout << "0"; cout << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "Массив обновлён. " << endl;

delete [] nm;

Take();

obaza.close();

}

void Udalit(){

ofstream obaza("baza.txt");

int nomer;

cout << "Номер удаляемого элемента: ";

cin >> nomer;

while (nomer<1||nomer>r||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать длину массива! Повторите ввод: ";

cin >> nomer;

}

r--;

obaza << "Количество элементов: " << r << endl;

Massiv \*nm = new Massiv [r];

for (int i=0;i<nomer-1;i++){

nm[i]=m[i];

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

for (int i=nomer;i<r+1;i++){

nm[i-1]=m[i];

obaza << i << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "Массив обновлён. " << endl;

delete [] nm;

Take();

}

void Set(){

const int fon = system("Color F0");

system("chcp 1251>nul");

setlocale(LC\_ALL, "Russian");

}

struct MS{

char sim;

int zns;

MS \*next;

};

void BouerMur(){

string kucha,isk;

int sdv,rms=1,poz,nom=0;

bool end=0,nai=0,sov=0;

for (int i=0;i<r;i++) kucha+=m[i].adres+'\n';

cout << "Адрес искомого элемента: " << endl;

cin.ignore(30,'\n');

getline (cin, isk);

MS \*first=new MS;

MS \*cur = first;

MS \*nov;

cur->next=NULL;

first->sim = isk[isk.length()-1];

first->zns = isk.length();

for (int i=isk.length()-2;i>=-1;i--){

cur = first;

nai=0;

while (cur->next!=NULL&&!nai){

if (isk[i]==cur->sim) nai=1;

cur=cur->next;

}

if (!nai){

nov = new MS;

nov->next=NULL;

nov->sim=isk[i];

nov->zns=rms++;

cur->next=nov;

cur=nov;

}

}

cur = first;

while (cur->next!=NULL) {

cout << "Символ: " << cur->sim << " Сдвиг: " << cur->zns << endl;

cur=cur->next;

}

poz=isk.length()-1;

nai=0;

cur=first;

int elm,kel;

bool tak=0;

while(poz<kucha.length()){

elm=isk.length()-1;

cur=first;

tak=0;

nom=0;

if (kucha[poz]==isk[elm]){

kel=poz;

while (kucha[kel]==isk[elm]&&elm>=0){

kel--;elm--;

}

if (elm==-1){sov=1;nai=1;}

}

for (int j=0;j<poz;j++) if (kucha[j]=='\n') nom++;

while (cur->next!=NULL&&!tak) {

if (kucha[poz]==cur->sim) tak=1;

else cur=cur->next;

}

if (tak){

if (poz+cur->zns<kucha.length()) poz+=cur->zns;

else poz=kucha.length();

}

else{

if (poz+isk.length()<kucha.length()) poz+=isk.length();

else poz=kucha.length();

}

if (sov){

cout << "Элемент, содержащий введённые данные: " << nom+1 << ". " << m[nom].fio << "; "; if (m[nom].d<10) cout << "0"; cout << m[nom].d << "."; if (m[nom].m<10) cout << "0"; cout << m[nom].m << "." << m[nom].g << "; " << m[nom].adres << ";" << endl;

}

sov=0;

}

if (!nai) cout << "Элемент не найден. " << endl;

}

void KMP(){

string kucha, isk;

for (int i=0;i<r;i++) kucha+=m[i].adres+'\n';

cout << kucha << endl; //!\_!

cout << "Адрес искомого элемента: "; cin.ignore(30,'\n'); getline(cin,isk);

int \*s = new int [isk.length()];

for (int i=1,j=0;i<isk.length();)

{

if (isk[i]==isk[j]) {s[i]=j+1;i++;j++;}

else {

if (j==0) s[i++]=0;

else j=s[j-1];

}

}

int i=0,j=0,nom=0;

bool nai=0;

while (i<kucha.length()){

nom=0;

if (kucha[i]==isk[j]){

i++;j++;

if (j==isk.length()){

for (int e=0;e<i;e++) if (kucha[e]=='\n') nom++;

cout << "Элемент, содержащий введённые данные: " << nom+1 << ". " << m[nom].fio << "; "; if (m[nom].d<10) cout << "0"; cout << m[nom].d << "."; if (m[nom].m<10) cout << "0"; cout << m[nom].m << "." << m[nom].g << "; " << m[nom].adres << ";" << endl;

nai=1;

}

}

else{

if (j==0) i++;

else j=s[j-1];

}

}

if (!nai) cout << "Элемент не найден. " << endl;

}

int main(){

Set();

setlocale(0,"");

while (v!=0){

cout << "Выберете действие: " << endl << "0. Завершить работу программы. \n1. Сформировать массив случайно; \n2. Сформировать массив самостоятельно; \n3. Скопировать массив из файла; \n4. Удалить элемент из массива; \n5. Добавить элемент в массив; ";

cout <<"\n6. Запустить метод поиска Боера-Мура; \n7. Запустить метод поиска Кнута-Морриса-Пратта. " << endl;

cin >> v;

switch(v){

case 0: return 0;

case 1: Create();break;

case 2: CUser(); break;

case 3: Take(); break;

case 4: if (mc==true) Udalit(); else cout << "База данных отсутствует. " << endl; break;

case 5: if (mc==true) Dop(); else cout << "База данных отсутствует. " << endl; break;

case 6: if (mc==true) BouerMur(); else cout << "База данных отсутствует. " << endl; break;

case 7: if (mc==true) KMP(); else cout << "База данных отсутствует. " << endl; break;

default: cout << "Неверный ввод! " << endl;

}

cout << endl;}}

**Результат работы программы**