Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №12 и 14**

Дисциплина: «Информатика»

Тема: Методы поиска: линейный поиск, интерполяционный поиск,

Прямой поиск подстроки в строке,

Метод Кнута-Морриса-Пратта, метод Бойера-Мура.

Вариант 14

Выполнил:

Студент группы ИВТ-20-1б

Кущ Данил Вячеславович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь 2021**

**Цель:** Поиск данных в динамических структурах по заданному ключу.

**Постановка задачи:**

1. Создать структуру и выполнить поиск различными методами по ключу соответствующему с вариантом.
2. Методы поиска: линейный поиск, интерполяционный поиск, прямой поиск подстроки в строке, метод Кнута-Морриса-Пратта, метод Бойера-Мура.

**Вариант:** 14  
Структура: ФИО, дата рождения, адрес.

Ключ: Адрес.

**Анализ задачи:**

1. Объявляем структуры в соответствии с нашим вариантом, затем создаем динамический массив структур на 100 элементов.
2. При помощи цикла while работающего, пока пользователь не захочет закончить программу, оформляем диалоговое окно с помощью switch, в котором вызываются разные функции в зависимости от того какое действие выбирает пользователь.
3. Функция ввода элементов в структуры из файла, через цикл for, ограниченный размером массива, записываем элементы на свои места.
4. Функция сохранения нашей базы в файл, где мы через цикл for записываем элементы в наш файл.
5. Функция вывода, в которой через цикл for выводятся все элементы.
6. Очистка консоли, для более удобного использования.
7. В случае ошибочного ввода выведется ошибка и запросит выбор пользователя еще раз.
8. Возможность выбрать поиск элемента по ключу, в этом блоке находится еще одно диалоговое окно switch, в котором спрашивается, какой из методов поиска вы хотите использовать.
9. Функция линейного поиска, в которой мы запрашиваем ключ и ищем элемент, в случае отсутствия элемента выведется соответствующая строка.
10. Функция прямого поиска подстроки в строке, в ней мы запрашиваем подстроку, то есть часть ключа. Проходим по всем элементам и выводим все те, которые подходят нам, если нет ни одного подходящего элемента, выводим соответствующую строку.
11. Функция интерполяционного поиска. Так как для данного поиска необходим отсортированный массив чисел, мы создаем временный дополнительный массив, для которого преобразуем все строки в числа путем суммирования всех символов по таблице ASCII. После чего полученный массив сортируем. Запрашиваем ключ у пользователя и преобразуем его в такую же сумму символов, по которой выполняем интерполяционный поиск. При нахождении элемента преобразуем его обратно и выводим пользователю, если элемента не обнаруживаем, выводим соответствующую строку.
12. Функция поиска Бойера-Мура. Запрашиваем подстроку (часть ключа) у пользователя. Через цикл for проходимся по всем элементам, через поиск Бойера-Мура проверяем подстроку со строкой, тем самым выводя все подходящие элементы. Если ни одного элемента не находится выводится строка об отсутствии подходящих элементов.
13. Функция метода поиска Кнута-Мориса-Пратта.Запрашиваем подстроку (часть ключа) у пользователя. Через цикл for проходимся по всем элементам, через поиск Кнута-Мориса-Пратта проверяем подстроку со строкой, тем самым выводя все подходящие элементы. Если ни одного элемента не находится выводится строка об отсутствии подходящих элементов.

**Код программы:**

|  |
| --- |
| #include <iostream> |
|  | #include <string> |
|  | #include <fstream> |
|  | using namespace std; |
|  |  |
|  | const int NotUsed = system( "color F0" ); |
|  | const int size = 100; |
|  |  |
|  | struct info{ |
|  | string FCs; |
|  | string birthday; |
|  | string address; |
|  | }; |
|  |  |
|  | info\* astr = new info[size]; |
|  |  |
|  | void make(){ |
|  | string a; |
|  | ifstream str("Struct.txt"); |
|  | for(int i=0; i<size; i++){ |
|  | if(str.is\_open()){ |
|  | if(getline(str, a)){ |
|  | astr[i].FCs = a; |
|  | } |
|  | if(getline(str, a)){ |
|  | astr[i].birthday = a; |
|  | } |
|  | if(getline(str, a)){ |
|  | astr[i].address = a; |
|  | } |
|  | } |
|  | } |
|  | str.close(); |
|  | } |
|  |  |
|  | void linear\_search(){ |
|  | string a; |
|  | bool n = true; |
|  | cout << "Enter the key: "; |
|  | getline(cin,a); |
|  | getline(cin,a); |
|  | for(int i=0; i<size; i++){ |
|  | if(astr[i].address == a){ |
|  | cout << "Element number: " << i+1 << endl; |
|  | cout << "This element: " << endl; |
|  | cout << " " << astr[i].FCs << endl; |
|  | cout << " " << astr[i].birthday << endl; |
|  | cout << " " << astr[i].address << endl; |
|  | n=false; |
|  | } |
|  | } |
|  | if(n=true){ |
|  | cout << "There is no element with this key" << endl; |
|  | } |
|  | } |
|  |  |
|  | void substr\_search(){ |
|  | string substr; |
|  | cout << "Enter the substring: "; |
|  | getline(cin,substr); |
|  | getline(cin,substr); |
|  | string str; |
|  | bool x=true, n=true; |
|  | for(int i=0; i<size; i++){ |
|  | str=astr[i].address; |
|  | for(int j=0; j<str.size(); j++){ |
|  | x=true; |
|  | for(int k=0; k<substr.size();k++){ |
|  | if(substr[k]!=str[j+k]){ |
|  | x=false; |
|  | } |
|  | } |
|  | if(x==true){ |
|  | cout << "Element number: " << i+1 << endl; |
|  | cout << "This element: " << endl; |
|  | cout << " " << astr[i].FCs << endl; |
|  | cout << " " << astr[i].birthday << endl; |
|  | cout << " " << astr[i].address << endl; |
|  | n=false; |
|  | } |
|  | } |
|  | } |
|  | if(n=true){ |
|  | cout << "There is no element with this key" << endl; |
|  | } |
|  | } |
|  |  |
|  |  |
|  | void interp\_search(){ |
|  | int\* arr = new int[size]; |
|  | int sum; |
|  | string str; |
|  | for(int i=0; i<size;i++){ |
|  | str=astr[i].address; |
|  | sum=0; |
|  | for(int j=0;j<str.size();j++){ |
|  | sum+=str[j]; |
|  | } |
|  | arr[i]=sum; |
|  | } |
|  | int min, temp; |
|  | for (int i = 0; i < size-1; i++) |
|  | { |
|  | min = i; |
|  | for (int j = i + 1; j < size; j++) |
|  | { |
|  | if (arr[j] < arr[min]){ |
|  | min = j; |
|  | } |
|  | } |
|  | temp = arr[i]; |
|  | arr[i] = arr[min]; |
|  | arr[min] = temp; |
|  | } |
|  | string key; |
|  | int sumkey; |
|  | cout << "Enter the key: "; |
|  | getline(cin,key); |
|  | getline(cin,key); |
|  | for(int i=0;i<key.size();i++){ |
|  | sumkey+=key[i]; |
|  | } |
|  | int mid, left=0, right = size-1; |
|  | int answer; |
|  | bool z=true; |
|  | while(arr[left]<=sumkey && arr[right]>= sumkey && z){ |
|  | mid = left + ((sumkey - arr[left]) \* (right - left))/(arr[right]-arr[left]); |
|  | if (arr[mid]<sumkey){ |
|  | left=mid+1; |
|  | }else{ |
|  | if(arr[mid]> sumkey){ |
|  | right = mid-1; |
|  | }else{ |
|  | answer=mid; |
|  | z=false; |
|  | } |
|  | } |
|  | } |
|  | sum=0; |
|  | int l=0; |
|  | if(z==true){ |
|  | if (arr[left]==sumkey){ |
|  | answer = left; |
|  | }else{ |
|  | answer = -1; |
|  | } |
|  | } |
|  | if (answer>-1){ |
|  | for(int i=0; i <size; i++){ |
|  | str=astr[i].address; |
|  | sum=0; |
|  | for(int j=0; j<str.size(); j++){ |
|  | sum+=str[j]; |
|  | } |
|  | if(sum==arr[answer]){ |
|  | cout << "Element number: " << i+1 << endl; |
|  | cout << "This element: " << endl; |
|  | cout << " " << astr[i].FCs << endl; |
|  | cout << " " << astr[i].birthday << endl; |
|  | cout << " " << astr[i].address << endl; |
|  | } |
|  | } |
|  | }else{ |
|  | cout << "There is no element with this key" << endl; |
|  | } |
|  | } |
|  |  |
|  | void boyer\_moore(){ |
|  | string ct; |
|  | cout << "Enter the substring: "; |
|  | getline(cin,ct); |
|  | getline(cin,ct); |
|  | string st; |
|  | int stl, ctl; |
|  | int answer; |
|  | bool m=true, n=true; |
|  | for(int z=0; z<size; z++){ |
|  | st=astr[z].address; |
|  | stl = st.size(); |
|  | ctl = ct.size(); |
|  | if (stl != 0 && ctl != 0 && stl >= ctl) |
|  | { |
|  | int i, p; |
|  | int b[256]; |
|  | for (i = 0; i < 256; i++) |
|  | { |
|  | b[i] = ctl; |
|  | } |
|  | for (i = ctl - 2; i >= 0; i--) |
|  | { |
|  | if (b[int((unsigned char)ct[i])] == ctl) |
|  | { |
|  | b[int((unsigned char)ct[i])] = ctl - i - 1; |
|  | } |
|  | } |
|  | p = ctl - 1; |
|  | n=true; |
|  | while (p < stl && n==true) |
|  | { |
|  | if (ct[ctl - 1] != st[p]) |
|  | { |
|  | p += b[int((unsigned char)st[p])]; |
|  | } |
|  | else |
|  | { |
|  | for (i = ctl - 1; i >= 0; i--) |
|  | { |
|  | if (ct[i] != st[p - ctl + i + 1]) |
|  | { |
|  | p += b[int((unsigned char)st[p])]; |
|  | break; |
|  | } |
|  | else if (i == 0) |
|  | { |
|  | cout << "Element number: " << z+1 << endl; |
|  | cout << "This element: " << endl; |
|  | cout << " " << astr[z].FCs << endl; |
|  | cout << " " << astr[z].birthday << endl; |
|  | cout << " " << astr[z].address << endl; |
|  | m=false; |
|  | n=false; |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | if(m==true){ |
|  | cout << "There is no element with this key" << endl; |
|  | } |
|  | } |
|  |  |
|  | bool KMP(string str,string key) |
|  | { |
|  | int stl = str.size(); |
|  | int ctl = key.size(); |
|  | if (stl != 0 && ctl != 0 && stl > ctl) |
|  | { |
|  | int i, j = 0, k = -1; |
|  | int\* pf = new int[1000]; |
|  | pf[0] = -1; |
|  | while (j < ctl - 1) |
|  | { |
|  | while (k >= 0 && key[j] != key[k]) |
|  | { |
|  | k = pf[k]; |
|  | } |
|  | j++; |
|  | k++; |
|  | if (key[j] == key[k]) |
|  | { |
|  | pf[j] = pf[k]; |
|  | } |
|  | else |
|  | { |
|  | pf[j] = k; |
|  | } |
|  | } |
|  | i = 0; |
|  | j = 0; |
|  | while (j < ctl && i < stl) |
|  | { |
|  | while (j >= 0 && str[i] != key[j]) |
|  | { |
|  | j = pf[j]; |
|  | } |
|  | i++; |
|  | j++; |
|  | } |
|  | delete[] pf; |
|  | if (j == ctl) |
|  | { |
|  | return true; |
|  | } |
|  | else |
|  | { |
|  | return false; |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | void print(){ |
|  | cout << "Your struct: " << endl; |
|  | for(int i=0; i<size;i++){ |
|  | cout << "Your " << i+1 << " person: " << endl; |
|  | cout << " " << astr[i].FCs << endl; |
|  | cout << " " << astr[i].birthday << endl; |
|  | cout << " " << astr[i].address << endl; |
|  | } |
|  | } |
|  |  |
|  |  |
|  | void save(){ |
|  | string a; |
|  | ofstream str; |
|  | str.open("Struct.txt"); |
|  | for(int i=0; i<size; i++){ |
|  | str << astr[i].FCs << endl; |
|  | str << astr[i].birthday << endl; |
|  | str << astr[i].address << endl; |
|  | } |
|  | str.close(); |
|  | } |
|  |  |
|  | int main() { |
|  | bool x=true; |
|  | int i; |
|  | string key; |
|  | bool m=true; |
|  | while(x){ |
|  | cout << "1 - Take info for structure from Struct.txt" << endl; |
|  | cout << "2 - Save structure in file Struct.txt" << endl; |
|  | cout << "3 - Print your structure" << endl; |
|  | cout << "4 - Find an element by a key Address" << endl; |
|  | cout << "9 - Clean the console" << endl; |
|  | cout << "0 - Exit the programm" << endl; |
|  | cout << "Your choice: "; |
|  | cin >> i; |
|  | switch(i){ |
|  | case 1: |
|  | make(); |
|  | break; |
|  | case 2: |
|  | save(); |
|  | break; |
|  | case 3: |
|  | print(); |
|  | break; |
|  | case 4: |
|  | int v; |
|  | cout << "1 - Linear search method" << endl; |
|  | cout << "2 - Method of straight search of substring in a string" << endl; |
|  | cout << "3 - Interpolation method" << endl; |
|  | cout << "4 - Boyer-Moore string-search" << endl; |
|  | cout << "5 - Knuth-Morris-Pratt string-search" << endl; |
|  | cout << "0 - Choose other operation" << endl; |
|  | cout << "Your choice: "; |
|  | cin >> v; |
|  | switch(v){ |
|  | case 1: |
|  | linear\_search(); |
|  | break; |
|  | case 2: |
|  | substr\_search(); |
|  | break; |
|  | case 3: |
|  | interp\_search(); |
|  | break; |
|  | case 4: |
|  | boyer\_moore(); |
|  | break; |
|  | case 5: |
|  | cout << "Enter the key: "; |
|  | getline(cin,key); |
|  | getline(cin,key); |
|  | for(int i=0;i<size;i++){ |
|  | if(KMP(astr[i].address, key)){ |
|  | cout << "Element number: " << i+1 << endl; |
|  | cout << "This element: " << endl; |
|  | cout << " " << astr[i].FCs << endl; |
|  | cout << " " << astr[i].birthday << endl; |
|  | cout << " " << astr[i].address << endl; |
|  | m=false; |
|  | } |
|  | } |
|  | if(m==true){ |
|  | cout << "There is no element with this key" << endl; |
|  | } |
|  | break; |
|  | case 0: |
|  | break; |
|  | default: |
|  | cout << "Unexcepted error, try again" << endl; |
|  | break; |
|  | } |
|  | break; |
|  | case 9: |
|  | system("cls"); |
|  | break; |
|  | case 0: |
|  | x = 0; |
|  | break; |
|  | default: |
|  | cout << "Unexcepted error, try again" << endl; |
|  | break; |
|  | } |
|  | } |
|  | return 0; |
|  | } |

**Результат работы программы:**













 

