ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | С. Ю. Гуков |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| №4.2. РАЗРАБОТКА АЛГОРИТМА ИСЧЕРПЫВАЮЩЕГО ПОИСКА (ПОЛНОГО ПЕРЕБОРА) |
| Вариант 2 (15) |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 5136 |  | , |  | Д. Л. Мягков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Оглавление:**

[Цель. 3](#_Toc96548818)

[Индивидуальное задание 4](#_Toc96548819)

[Описание алгоритма 5](#_Toc96548821)

[Листинг программы. 6](#_Toc96548823)

[Результат. 13](#_Toc96548824)

[Вывод. 15](#_Toc96548825)

# Цель.

Реализовать алгоритм на языке C/С++, выполняющий поставленную задачу (АЛГОРИТМ ДОЛЖЕН БЫТЬ РЕАЛИЗОВАН ПУТЕМ ПЕРЕБОРА ВСЕХ ВОЗМОЖНЫХ ВАРИАНТОВ РЕШЕНИЯ). Вариант задания, пример входных и выходных данных представлен в таблице 9. Глобальные параметры использовать запрещено; допустимо использование дополнительных функций.

* Разработанный алгоритм должен быть реализован в виде цельной программной функции (или нескольких функций) так, чтобы мог быть многократно применим с различными исходными данными и при этом не включал команды, не относящиеся к решаемой задаче, например, ввод и вывод исходных данных на консоль или в файл.
* Произвести экспериментальную проверку времени работы разработанного алгоритма, определив его класс сложности для среднего случая. Измерить среднее время для Test\_Count повторений при различных размерностях входных данных. !ВАЖНО: при относительной небольшой размерности входных данных программа может потребовать для выполнения значительное время, рекомендуется не ждать выполнения программы в случае превышения времени выполнения 15 минут.

# Индивидуальное задание.

## 

Рисунок 1. Индивидуальное задание

# Описание алгоритма

# Листинг программы.

Листинг lab\_2\_2.cpp:

#include <iostream>// библиотека

#include <iomanip>

#include <Windows.h>

#include "various\_functions.h"

#include "main\_header.h"

#include <vector>

#include "string"

using namespace std;

int binary\_find(int key, vector<vector<int>>& array, int left, int right);

void sorting\_bubble\_two\_dimensional\_vectors(vector<vector<int>>& array);

void in\_vector(vector <string>& As);

void fill\_Hs(vector<vector<int>>& Hs, vector<string> As);

void print\_Hs(vector<vector<int>>& Hs);

int input\_P();

int input\_M();

string input\_Is();

int input\_value(int a);

void lab\_2\_2()

{

setlocale(0, ""); // поддержка кириллицы в консоли (вывод)

SetConsoleCP(1251); // поддержка кириллицы в консоли (ввод)

SetConsoleOutputCP(1251); // поддержка кириллицы в консоли (ввод)

vector <string> As;

in\_vector(As);

string Is = input\_Is();

As.push\_back(Is);

vector <vector <int>> Hs(As.size(), vector <int> (2));

fill\_Hs(Hs, As);

int hash\_Is = Hs[Hs.size() - 1][0];

cout << "Hash Is: " << hash\_Is << endl;

Hs.erase(Hs.begin() + Hs.size() - 1);

print\_Hs(Hs);

sorting\_bubble\_two\_dimensional\_vectors(Hs);

cout << "Ias: " << binary\_find(hash\_Is, Hs, 0, Hs.size() - 1);

}

void sorting\_bubble\_two\_dimensional\_vectors(vector<vector<int>>& Hs)

{

for (int i = 0; i < Hs.size(); i++)

for (int j = 0; j < Hs.size() - 1; j++)

if (Hs[j] > Hs[j + 1])

swap\_notstd(Hs[j], Hs[j + 1]);

}

string input\_Is()

{

cout << "Введите строку Is: ";

string Is;

getline(cin, Is);

return Is;

}

void print\_Hs(vector<vector<int>>& Hs)

{

cout << "Hs: \n";

for (int i = 0; i < Hs.size(); i++)

cout << setw(2) << i << ": " << Hs[i][0] << endl;

}

int input\_P()

{

cout << "Введите P: ";

return input\_value(10);

}

int input\_M()

{

cout << "Введите M: ";

return input\_value(10001);

}

void in\_vector(vector <string>& As)

{

srand(clock());

cout << "Введите длину массива: ";

int N;

string str;

getline(cin, str);

if (str == "")

{

N = rand() % 10 + 10;

}

else N = get\_number\_int\_without\_cin(str);

cout << "Длина массива = " << N << endl;

int size\_str;

for (int i = 0; i < N; i++)

{

size\_str = rand() % 5 + 10;

str = "";

for (int j = 0; j < size\_str; j++)

{

str += rand() % (10) + 'A';

}

cout << setw(2) << i << ": " << str << endl;

As.push\_back(str);

}

}

void fill\_Hs(vector<vector<int>>& Hs, vector<string> As)

{

unsigned long long H = 0;

int P = input\_P();

//cout << "Введите P: ";

//int P = get\_number\_int();

int M = input\_M();

//cout << "Введите M: ";

//int M = get\_number\_int();

for (int i = 0; i < Hs.size(); i++)

{

for (int j = 0; j < As[i].length(); j++)

{

H += (As[i][j] - 'A') \* pow(P, j);

}

Hs[i][0] = H % M;

Hs[i][1] = i;

H = 0;

}

}

int binary\_find(int key, vector<vector<int>>& array, int left, int right)

{

int mid = (left + right) / 2;

if (key == array[mid][0])

{

return array[mid][1];

}

else if (left > right)

{

return -1;

}

else if (key > array[mid][0])

{

left = mid + 1;

return binary\_find(key, array, left, right);

}

else

{

right = mid - 1;

return binary\_find(key, array, left, right);

}

}

Листинг main.cpp:

#include <iostream>// библиотека

#include <Windows.h> // библиотека

#include "main\_header.h"

#include "various\_functions.h"

int main()

{

setlocale(0, ""); // поддержка кириллицы в консоли (вывод)

SetConsoleCP(1251); // поддержка кириллицы в консоли (ввод)

SetConsoleOutputCP(1251); // поддержка кириллицы в консоли (ввод)

double task\_number;

boolean debugging = true;

if (debugging == true)

{

//lab\_2\_1();

//lab\_4\_2();

dop\_task\_1();

}

else

{

std::cout

<< "Готовые лабораторные работы: " << std::endl

<< std::endl

<< " 1.1 Поиск макс. суммы двух элементов " << std::endl

<< " 1.2 Цикл с вычислением функции " << std::endl

<< " 1.3 Поиск неповторяющихся остатков от деления " << std::endl

<< " 2.1 Количество скобок " << std::endl

<< " 2.2 Бинарный поиск строки по хэшу " << std::endl

<< " 3.1 Рекурсия " << std::endl

<< " 4.1 Оценка алгоритма (Гаусс) " << std::endl

<< " 4.2 База данных " << std::endl

<< " 5.1 Битовая последовательность " << std::endl

<< std::endl

<< "Введите номер лабораторной работы: ";

while (true)

{

task\_number = get\_number\_double();

if (task\_number == 1.1 ||

task\_number == 1.2 ||

task\_number == 1.3 ||

task\_number == 2.1 ||

task\_number == 2.2 ||

task\_number == 3.1 ||

task\_number == 4.1 ||

task\_number == 4.2 ||

task\_number == 5.1)

{

system("cls");

std::cout << "Лабораторная работа № " << task\_number << std::endl;

break;

}

else std::cout << "Такой лабораторной работы нет, введите другой номер: ";

}

if (task\_number == 1.1)

{

lab\_1\_1();

}

else if (task\_number == 1.2)

{

lab\_1\_2();

}

else if (task\_number == 1.3)

{

lab\_1\_3();

}

else if (task\_number == 2.1)

{

lab\_2\_1();

}

else if (task\_number == 2.2)

{

lab\_2\_2();

}

else if (task\_number == 3.1)

{

lab\_3\_1();

}

else if (task\_number == 4.1)

{

lab\_4\_1();

}

else if (task\_number == 4.2)

{

lab\_4\_2();

}

else if (task\_number == 5.1)

{

lab\_5\_1();

}

}

return 0;

}

Листинг various\_functions.h:

#include <string>

double get\_number\_double();

int get\_number\_int();

int get\_number\_int\_without\_cin(std::string str);

int input\_value(int a);

template <typename T>

T str\_to\_number(std::string str)

{

T a = 0;

int k = 0, m = 1;

if (str[0] == '-')

{

m = -1;

str.erase(0, 1);

}

if (str.find('.') != -1)

{

k = str.find('.');

str.erase(str.find('.'), 1);

}

else k = str.length();

for (int i = 0; i < str.length(); i++)

{

a = a \* 10 + (str[i] - '0');

}

return m \* (a / pow(10, (str.length() - k)));

}

Листинг various\_functions.cpp:

#include <iostream>// библиотека

#include <Windows.h> // библиотека

#include "various\_functions.h"

#include <vector>

using namespace std;

template <typename T>

T str\_to\_number(std::string str);

double get\_number\_double()

{

while (true) // цикл продолжается до тех пор, пока пользователь не введет корректное значение

{

std::string str;

std::cin >> str;

int k = 1;

double a;

if (str.length() > 1)

{

if (((str[0] == '-') && (str[1] == '.' ||

(str[1] == '0' && str[2] != '.'))) ||

str[0] == '.' || (str[0] == '0' && str[1] != '.' && str[1] != '\n'))

{

k \*= 0;

std::cout << "!!Вы ввели не число!!" << std::endl << "Введитe число: ";

}

}

else

{

for (int i = 0; i < str.length(); i++)

{

if ((str[i] < '0' || str[i] > '9') && str[i] != '.' && str[i] != '-')

{

k \*= 0;

std::cout << "!!Вы ввели не число!!" << std::endl << "Введите число: ";

break;

}

}

}

if (k == 1)

{

a = str\_to\_number<double>(str);

return a;

break;

}

}

}

Результат.

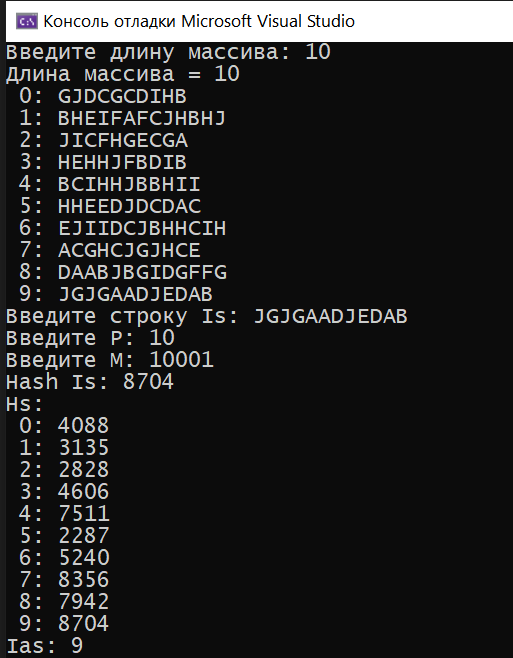


Рисунок 2. Тест

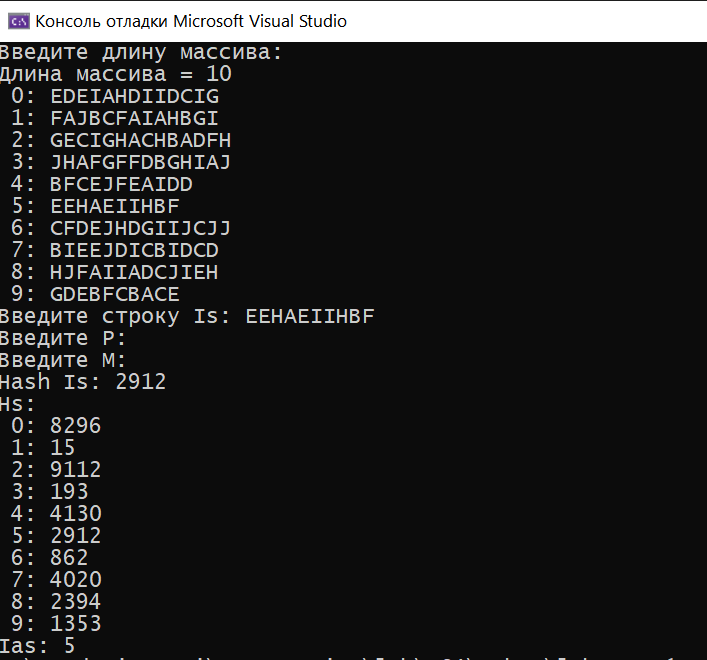


Рисунок 3. Тест

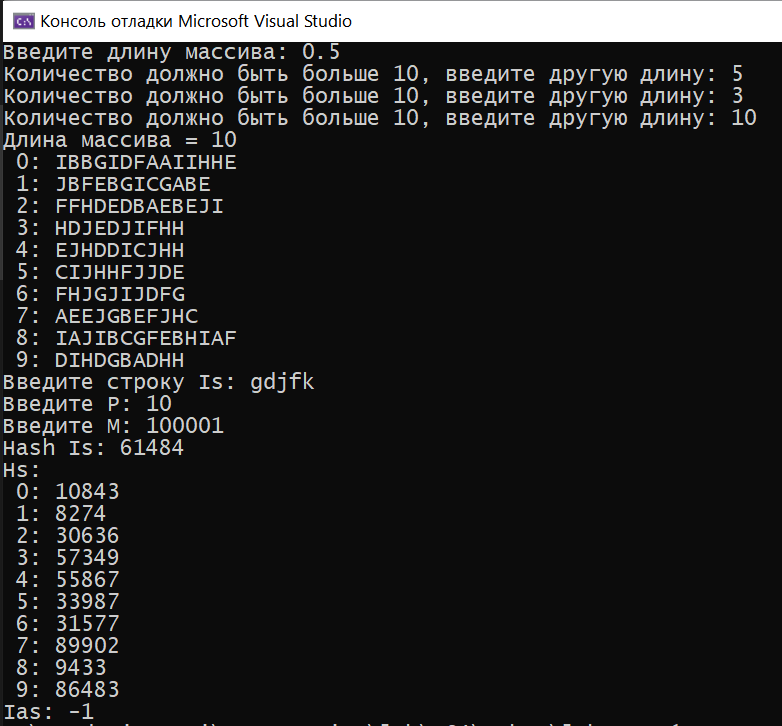


Рисунок 4. Тест

# Вывод.

В ходе выполнения лабораторной работы были освоены:

* векторы
* получение хэшей строк
* алгоритм бинарного поиска

Вводится массив строк. Вычисляется массив хэшей. Вводится строка. С помощью бинарного поиска находится и выводится индекс введённой строки в исходном массиве строк.