**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №4**

**по дисциплине «Программирование»**

Тема: Текстовые строки как массивы символов

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. | Чекстер А. А. |  |
| Преподаватель | Глущенко А. Г. |  |

Санкт-Петербург

2022

**Цель работы.**

Изучение представления текстовых строк в программах. Работа со строками как с массивами символов. Операции над строками. Знакомство с классом **std::string**.

**Основные теоретические положения.**

Текстовые строки представляются с помощью одномерных массивов символов. В языке **C++** текстовая строка представляет собой набор символов, обязательно заканчивающийся нулевым символом (‘\0’). Поэтому, если вы хотите создать текстовый массив для хранения 10 (N) символов, нужно выделить память под 11 (N+1) символов.

Объявленный таким образом массив может использоваться для хранения текстовых строк, содержащих не более 10 символов. Нулевой символ позволяет определить границу между содержащимся в строке текстом и неиспользованной частью строки.

Класс string предназначен для работы со строками типа char, которые представляют собой строчку с завершающим нулем (символ ‘\0’). Класс string был введен как альтернативный вариант для работы со строками типа char.

Чтобы использовать возможности класса **string**, нужно подключить библиотеку <string> и пространство имен **std**. Объявление же переменной типа **string** осуществляется схоже с обычной переменной:

**Постановка задачи.**

Необходимо написать программу, которая:

1) С клавиатуры или с файла (пользователь сам может выбрать способ ввода) вводится последовательность, содержащая от 1 до 50 слов, в каждом из которых от 1 до 10 строчных латинских букв и цифр. Между соседними словами произвольное количество пробелов. За последним символом стоит точка.

2) Необходимо отредактировать входной текст:

- удалить лишние пробелы;

- удалить лишние знаки препинания (под «лишними» подразумевается несколько подряд идущих знаков (обратите внимание, что «…» - корректное использование знака) в тексте);

- исправить регистр букв, если это требуется (пример некорректного использования регистра букв: пРиМЕр);

3) Выполнить задание по варианту: вывести на экран слова последовательности в обратном порядке.

4) Выполнить задание по варианту: вывести на экран ту же последовательность, заменив во всех словах первую букву соответствующей прописной буквой.

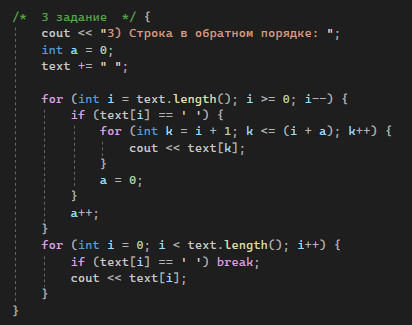
5) Необходимо найти все подстроки, которую введёт пользователь в имеющейся строке. Реализуйте два алгоритма: первый алгоритма – Линейный поиск, а второй алгоритм согласно вашему номеру в списке. Четные номера должны реализовать алгоритм КНМ, а нечетные – Бойера-Мура.

**Выполнение работы.**

Код программы представлен в приложении А.

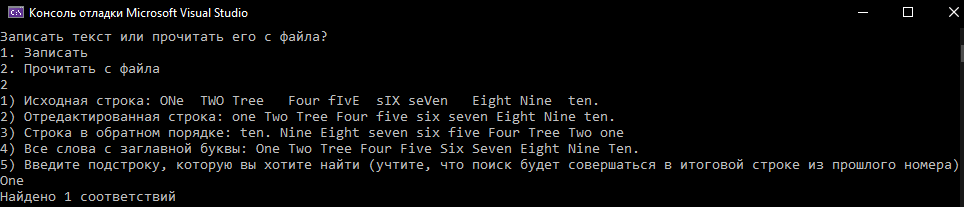
**Блок описания кода и использованных алгоритмов:**

1. Пользователь вводит строку или считывает её с файла.
2. Эта строка редактируется с помощью прописанных функций: 1 - удаляет лишние пробелы, 2 - удаляет лишние знаки препинания. 3 – исправляет регистры букв (если первая буква заглавная, то считается, что она и должна такой быть).
3. Слова выводятся в обратном порядке согласно программе:



1. В уже отредактированной строке во всех словах первая буква заменяется на заглавную. Делается это с помощью соответствующей функции.
2. Линейным поиском находится подстрока введённая пользователем.

**Блок скриншотов работы программы**



**Выводы.**

Во время выполнения данной работы были изучены строки, массивы символов и принципы работы с ними. Также был получен навык работы со строками.

Приложение А

рабочий код

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <chrono>

using namespace std;

using namespace chrono;

//функция для сортировки - 5

void quicksort(int\* randArr, int end, int begin)

{

//auto start = chrono::steady\_clock::now();

int mid,swap;

int f = begin;

int l = end;

mid = randArr[(f + l) / 2];

while (f < l)

{

while (randArr[f] < mid) f++;

while (randArr[l] > mid) l--;

if (f <= l)

{

swap = randArr[f];

randArr[f] = randArr[l];

randArr[l] = swap;

f++;

l--;

}

}

if (begin < l) quicksort(randArr, l, begin);

if (f < end) quicksort(randArr, end, f);

/\*auto end = chrono::steady\_clock::now();

cout << "Elapsed time in seconds: " << chrono::duration\_cast<chrono::nanoseconds>(end - start).count() << " nanosec" << endl;\*/

}

int Binarny(int arr[], int left, int right, int key)

{

int midd = 0;

while (1)

{

midd = (left + right) / 2;

if (key < arr[midd]) // если искомое меньше значения в ячейке

right = midd - 1; // смещаем правую границу поиска

else if (key > arr[midd]) // если искомое больше значения в ячейке

left = midd + 1; // смещаем левую границу поиска

else // иначе (значения равны)

return midd; // функция возвращает индекс ячейки

if (left > right) // если границы сомкнулись

return -1;

}

}

int main()

{

int snachala;

while (true) {

setlocale(LC\_ALL, "rus");

int randArr[100]{};

int first, second;

srand(time(NULL));

for (int i = 0; i < 100; i++)

{

randArr[i] = rand() % 189 - 99; // -99 ... 99

}

cout << endl << "Массив c числами oт -99 до 99: ";

for (int i = 0; i < 100; i++)

{

cout << randArr[i] << " ";

}

cout << endl;

int sort\_num;

cout << "Введите номер способа сортировки: " << endl;

cout << "1. Bubble sort (пузырьковая сортировка)" << endl;

cout << "2. Shaker sort (шейкер-сортировка)" << endl;

cout << "3. Comb sort (сортировка расчёской)" << endl;

cout << "4. Insert sort(сортировка вставками)" << endl;

cout << "5. Quick sort (быстрая сортировка)" << endl;

cin >> sort\_num;

// 1 - Bubble sort (пузырьковая сортировка)

if (sort\_num == 1) {

int key = 0;

int index = 0;

auto start = chrono::steady\_clock::now();

int const n = 100;

int k = 0;

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

for (k = 0; k < n - j - 1; ++k)

if (randArr[k] > randArr[k + 1]) {

double x = randArr[k];

randArr[k] = randArr[k + 1];

randArr[k + 1] = x;

}

}

for (int o = 0; o < n; o++)

cout << randArr[o] << ' ';

cout << endl;

cout << "\n\nВведите любое число для бинарного поиска: ";

cin >> key;

index = Binarny(randArr, 0, n, key);

if (index >= 0)

cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";

else

cout << "В массиве нет такого числа!\n\n";

cout << randArr[99] << ' ' << randArr[0] << endl;

float middle\_num;

middle\_num = (randArr[0] + randArr[99]) / 2;

if (int(middle\_num \* 100) % 100 > 50)

middle\_num = int(middle\_num) + 1;

else

middle\_num = int(middle\_num);

cout << middle\_num << " ";

int count\_i = 0;

for (int d = 0; d < 100; d++) {

if (randArr[d] == middle\_num) {

cout << d << " ";

count\_i += 1;

}

}

cout << " ";

cout << count\_i << endl;

auto end = chrono::steady\_clock::now();

cout << "Elapsed time in seconds: " << chrono::duration\_cast<chrono::nanoseconds>(end - start).count() << " nanosec" << endl;

int a, b;

cout << "Введите 2 числа для вывода значений из массива, которые: " << endl;

cout << "1.меньше 1-го числа " << endl;

cout << "2.больше 2-го числа: " << endl;

cin >> a >> b;

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

if (randArr[l] < a)

cout << randArr[l] << ' ';

}

cout << endl;

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

if (randArr[p] > b)

cout << randArr[p] << ' ';

}

cout << endl;

int q, w, t, z;

cout << "Введите 2 числа для перемещения значений чисел, индексы которых вы вводите: " << endl;

cin >> q >> w;

z = randArr[w];

t = randArr[q];

for (int u = 0; u < 100; ++u) {

if (u == q) {

randArr[q] = z;

//cout << z << " ";

}

if (u == w) {

randArr[w] = t;

cout << t << " ";

}

else

cout << randArr[u] << " ";

}

}

// 2 - Shaker sort (шейкер-сортировка)

if (sort\_num == 2) {

int key = 0;

int index = 0;

auto start = chrono::steady\_clock::now();

int const n = 100;

int c = 0;

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

for (int k = 0; k < n - j - 1; ++k)

if (randArr[k] > randArr[k + 1]) {

double x = randArr[k];

randArr[k] = randArr[k + 1];

randArr[k + 1] = x;

c += 1;

}

}

if (c >= n - 1) {

for (int j = n - 1; j > 0; --j) {

for (int k = n - 1; k > 1; --k)

if (randArr[k] > randArr[k + 1]) {

double x = randArr[k];

randArr[k] = randArr[k + 1];

randArr[k + 1] = x;

c += 1;

}

}

}

for (int o = 0; o < n; ++o)

if (o != 2)

cout << randArr[o] << ' ';

cout << endl;

cout << "\n\nВведите любое число для бинарного поиска: ";

cin >> key;

index = Binarny(randArr, 0, n, key);

if (index >= 0)

cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";

else

cout << "В массиве нет такого числа!\n\n";

cout << randArr[99] << ' ' << randArr[0] << endl;

float middle\_num;

middle\_num = (randArr[0] + randArr[99]) / 2;

if (int(middle\_num \* 100) % 100 > 50)

middle\_num = int(middle\_num) + 1;

else

middle\_num = int(middle\_num);

cout << middle\_num << " ";

int count\_i = 0;

for (int d = 0; d < 100; d++) {

if (randArr[d] == middle\_num) {

cout << d << " ";

count\_i += 1;

}

}

cout << " ";

cout << count\_i << endl;

auto end = chrono::steady\_clock::now();

cout << "Elapsed time in seconds: " << chrono::duration\_cast<chrono::nanoseconds>(end - start).count() << " nanosec" << endl;

int a, b;

cout << "Введите 2 числа для вывода значений из массива, которые: " << endl;

cout << "1.меньше 1-го числа " << endl;

cout << "2.больше 2-го числа: " << endl;

cin >> a >> b;

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

if (randArr[l] < a)

cout << randArr[l] << ' ';

}

cout << endl;

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

if (randArr[p] > b)

cout << randArr[p] << ' ';

}

cout << endl;

int q, w, t, z;

cout << "Введите 2 числа для перемещения значений чисел, индексы которых вы вводите: " << endl;

cin >> q >> w;

z = randArr[w];

t = randArr[q];

for (int u = 0; u < 100; ++u) {

if (u == q) {

randArr[q] = z;

//cout << z << " ";

}

if (u == w) {

randArr[w] = t;

cout << t << " ";

}

else

cout << randArr[u] << " ";

}

}

// 3 - Comb sort (сортировка расчёской)

if (sort\_num == 3) {

int key = 0;

int index = 0;

auto start = chrono::steady\_clock::now();

int swap;

int const n = 100;

float k = 1.247, S = n - 1;

int count = 0;

while (S >= 1)

{

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

{

if (randArr[i] > randArr[int(i + S)])

{

swap = randArr[int(i + S)];

randArr[int(i + S)] = randArr[i];

randArr[i] = swap;

}

}

S /= k;

}

while (true)

{

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

{

if (randArr[i] > randArr[i + 1])

{

swap = randArr[i + 1];

randArr[i + 1] = randArr[i];

randArr[i] = swap;

}

else count++;

}

if (count == n - 1)

break;

else

count = 0;

}

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

{

cout << randArr[i] << " ";

}

cout << endl;

cout << "\n\nВведите любое число для бинарного поиска: ";

cin >> key;

index = Binarny(randArr, 0, n, key);

if (index >= 0)

cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";

else

cout << "В массиве нет такого числа!\n\n";

for (int o = 0; o < n; o++)

cout << randArr[o] << ' ';

cout << endl;

cout << randArr[99] << ' ' << randArr[0] << endl;

float middle\_num;

middle\_num = (randArr[0] + randArr[99]) / 2;

if (int(middle\_num \* 100) % 100 > 50)

middle\_num = int(middle\_num) + 1;

else

middle\_num = int(middle\_num);

cout << middle\_num << " ";

int count\_i = 0;

for (int d = 0; d < 100; d++) {

if (randArr[d] == middle\_num) {

cout << d << " ";

count\_i += 1;

}

}

cout << " ";

cout << count\_i << endl;

auto end = chrono::steady\_clock::now();

cout << "Elapsed time in seconds: " << chrono::duration\_cast<chrono::nanoseconds>(end - start).count() << " nanosec" << endl;

int a, b;

cout << "Введите 2 числа для вывода значений из массива, которые: " << endl;

cout << "1.меньше 1-го числа " << endl;

cout << "2.больше 2-го числа: " << endl;

cin >> a >> b;

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

if (randArr[l] < a)

cout << randArr[l] << ' ';

}

cout << endl;

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

if (randArr[p] > b)

cout << randArr[p] << ' ';

}

cout << endl;

int q, w, t, z;

cout << "Введите 2 числа для перемещения значений чисел, индексы которых вы вводите: " << endl;

cin >> q >> w;

z = randArr[w];

t = randArr[q];

for (int u = 0; u < 100; ++u) {

if (u == q) {

randArr[q] = z;

}

if (u == w) {

randArr[w] = t;

cout << t << " ";

}

else

cout << randArr[u] << " ";

}

cout << endl;

}

// 4 - Insert sort (сортировка вставками)

if (sort\_num == 4) {

int key = 0;

int index = 0;

auto start = chrono::steady\_clock::now();

int const n = 100;

int k = 0, c = 0;

for (int j = 1; j < n; ++j) {

c = randArr[j];

for (k = j - 1; k >= 0 && randArr[k] > c; k--)

randArr[k + 1] = randArr[k];

randArr[k + 1] = c;

}

auto end = chrono::steady\_clock::now();

cout << "Elapsed time in seconds: " << chrono::duration\_cast <chrono::nanoseconds>(end - start).count() << " nanosec" << endl;

for (int o = 0; o < n; o++)

cout << randArr[o] << " ";

cout << endl;

cout << "\n\nВведите любое число для бинарного поиска: ";

cin >> key;

index = Binarny(randArr, 0, n, key);

if (index >= 0)

cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";

else

cout << "В массиве нет такого числа!\n\n";

cout << randArr[0] << ' ' << randArr[99] << endl;

float middle\_num;

middle\_num = (randArr[0] + randArr[99]) / 2;

if (int(middle\_num \* 100) % 100 > 50)

middle\_num = int(middle\_num) + 1;

else

middle\_num = int(middle\_num);

cout << middle\_num << " ";

int count\_i = 0;

for (int d = 0; d < 100; d++) {

if (randArr[d] == middle\_num) {

cout << d << " ";

count\_i += 1;

}

}

cout << " ";

cout << count\_i << endl;

int a, b;

cout << "Введите 2 числа для вывода значений из массива, которые: " << endl;

cout << "1.меньше 1-го числа " << endl;

cout << "2.больше 2-го числа: " << endl;

cin >> a >> b;

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

if (randArr[l] < a)

cout << randArr[l] << ' ';

}

cout << endl;

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

if (randArr[p] > b)

cout << randArr[p] << ' ';

}

cout << endl;

int q, w, t, z;

cout << "Введите 2 числа для перемещения значений чисел, индексы которых вы вводите: " << endl;

cin >> q >> w;

z = randArr[w];

t = randArr[q];

for (int u = 0; u < 100; ++u) {

if (u == q) {

randArr[q] = z;

}

if (u == w) {

randArr[w] = t;

cout << t << " ";

}

else

cout << randArr[u] << " ";

}

cout << endl;

return 0;

}

// 5 - Quick sort (быстрая сортировка)

if (sort\_num == 5) {

int key = 0;

int index = 0;

const int n = 100;

int end = n - 1, begin = 0;

quicksort(randArr, end, begin);

cout << endl << "Отсортированный массив: ";

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

cout << randArr[i] << " ";

for (int o = 0; o < n; o++)

cout << randArr[o] << ' ';

cout << endl;

cout << "\n\nВведите любое число для бинарного поиска: ";

cin >> key;

index = Binarny(randArr, 0, n, key);

if (index >= 0)

cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";

else

cout << "В массиве нет такого числа!\n\n";

cout << randArr[99] << ' ' << randArr[0] << endl;

float middle\_num;

middle\_num = (randArr[0] + randArr[99]) / 2;

if (int(middle\_num \* 100) % 100 > 50)

middle\_num = int(middle\_num) + 1;

else

middle\_num = int(middle\_num);

cout << middle\_num << " ";

int count\_i = 0;

for (int d = 0; d < 100; d++) {

if (randArr[d] == middle\_num) {

cout << d << " ";

count\_i += 1;

}

}

cout << " ";

cout << count\_i << endl;

int a, b;

cout << "Введите 2 числа для вывода значений из массива, которые: " << endl;

cout << "1.меньше 1-го числа " << endl;

cout << "2.больше 2-го числа: " << endl;

cin >> a >> b;

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

if (randArr[l] < a)

cout << randArr[l] << ' ';

}

cout << endl;

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

if (randArr[p] > b)

cout << randArr[p] << ' ';

}

cout << endl;

int q, w, t, z;

cout << "Введите 2 числа для перемещения значений чисел, индексы которых вы вводите: " << endl;

cin >> q >> w;

z = randArr[w];

t = randArr[q];

for (int u = 0; u < 100; ++u) {

if (u == q) {

randArr[q] = z;

}

if (u == w) {

randArr[w] = t;

cout << t << " ";

}

else

cout << randArr[u] << " ";

}

cout << endl;

}

cout << "\n\nХотите продолжить? Если Да, введите: '0'" << endl;

cin >> snachala;

if (snachala != 0)

break;

}

}