Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 16

По дисциплине «Основы алгоритмизации и программирования»

На тему «Функции пользователя»

Выполнил:

Студент 1 курса 9 группы

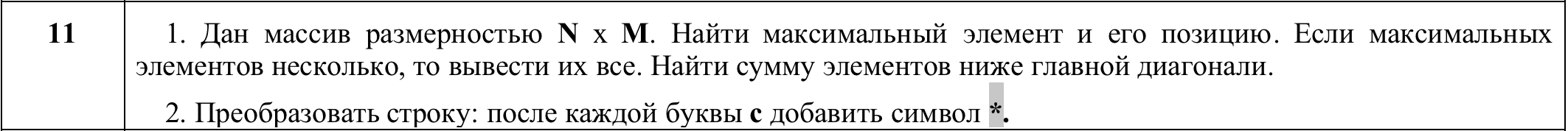
Павлович Ян Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 11

4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.



1. Дан массив размерностью **N** x **M**. Найти максимальный элемент и его позицию. Если максимальных элементов несколько, то вывести их все. Найти сумму элементов ниже главной диагонали.

#include <iostream> //для работы с вводом/выводом

**using** **namespace** std; //объявляем программе, что использем пространство имен std

//функция для поиска максимальных элементов в массиве

**void** find\_max(**int**\*\* arr, **int** n, **int** m) {

**int** max = arr[0][0]; //первый элемент массива обозначаем максимальным

    //с помощью циклов и сравнения ищем максимальный элемент

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

**for** (**int** j = 0; j < m; j++) {

**if** (max < arr[i][j]) {

                max = arr[i][j];

            }

        }

    }

    //с помощью циклов и сравнения ищем элементы, равные максимальному

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

**for** (**int** j = 0; j < m; j++) {

**if** (max == arr[i][j]) {

                cout << i + 1 << " " << j + 1 << ", ";

            }

        }

    }

}

//функция для вычесления суммы элементов под главноей диагональю

**int** dw\_sum(**int**\*\* arr, **int** n, **int** m) {

**int** sum = 0;

**for** (**int** i = 1; i < n; i++) {

**for** (**int** j = 0; j < i; j++) {

            sum += arr[i][j];

        }

    }

**return** sum;

}

**int** main() {

    setlocale(LC\_ALL, "rus"); //добавляем возможность работы с русским языком

**int**\*\* arr;

**int** n, m;

    cout << "Введите размер массива: "; cin >> n >> m;

    cout << "Введите массив" << endl;

    //создаем массив

    arr = **new** **int**\* [n];

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

        arr[i] = **new** **int**[m];

    }

    //вводим элементы в массив

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

**for** (**int** j = 0; j < m; j++) {

            cin >> arr[i][j];

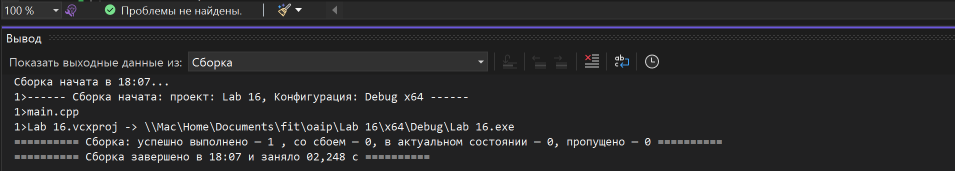
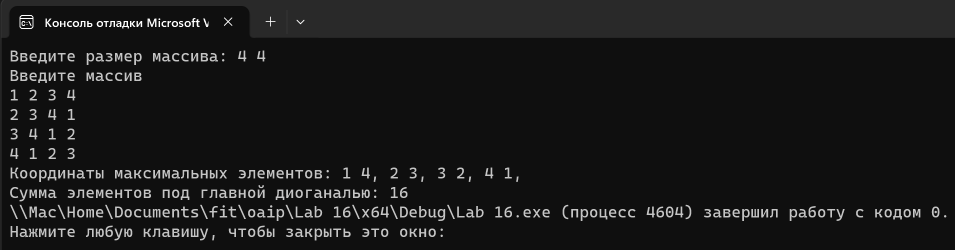
        }

    }

    cout << "Координаты максимальных элементов: "; find\_max(arr, n, m); cout << endl;

    cout << "Сумма элементов под главной диоганалью: " << dw\_sum(arr, n, m);

}



1. Преобразовать строку: после каждой буквы **c** добавить символ **\*.**

#include <iostream> //для работы с вводом/выводом

#include <Windows.h> //для работы с опирационной системой windows

#include <string> //для работы со строками

**using** **namespace** std; //объявляем программе, что использем пространство имен std

//функция для вывода строки с изменениями

**void** find(string s)  {

    //пока i меньше длины строки ищем c, чтобы добавить \*

**for** (**int** i = 0; i < s.length(); i++) {

        cout << s[i];

**if** (s[i] == 'с') {

            cout << "\*";

        }

    }

}

**int** main() {

    SetConsoleCP(1251); //установка кодировки

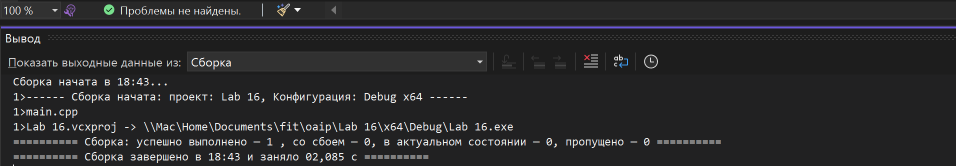
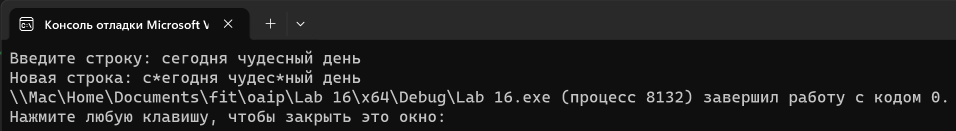
    setlocale(LC\_ALL, "rus"); //добопляем возможность работы с русским языком

    string s; //создаем строку

    cout << "Введите строку: "; getline(cin, s); //ввод строки

    cout << "Новая строка: "; find(s);

}

****

**Дополнительные задачи**

Вариант 7

4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.

****

1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов, а также число **d**. Найти строку, содержащую число **d**.

#include <iostream> //для работы с вводом/выводом

**using** **namespace** std; //объявляем программе, что использем пространство имен std

//функфия для поиска строки, которая содержит d

**int** find(**int**\*\* a, **int** d) {

**for** (**int** i = 0; i < **sizeof**(a[0]); i++) {

**for** (**int** j = 0; j < **sizeof**(a[0][0]); j++) {

**if** (a[i][j] == d) {

**return** i + 1;

            }

        }

    }

}

**int** main() {

    setlocale(LC\_ALL, "rus"); //добовляем возможность работы с русским языком

**int** m, n, d;

    cout << "Введите размер массива: "; cin >> n >> m;

    cout << "Введите d: "; cin >> d;

    //выделяем память для массива

    cout << "Введите массив: \n";

**int**\*\* a;

    a = **new** **int**\* [n];

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

        a[i] = **new** **int**[m];

    }

    //вводим элементы в массив

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

**for** (**int** j = 0; j < m; j++) {

            cin >> a[i][j];

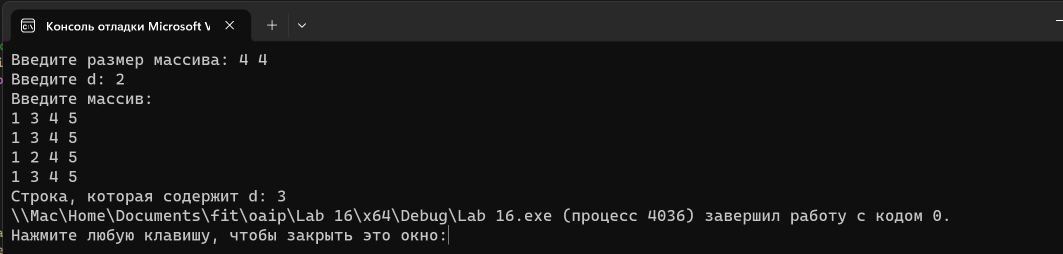
        }

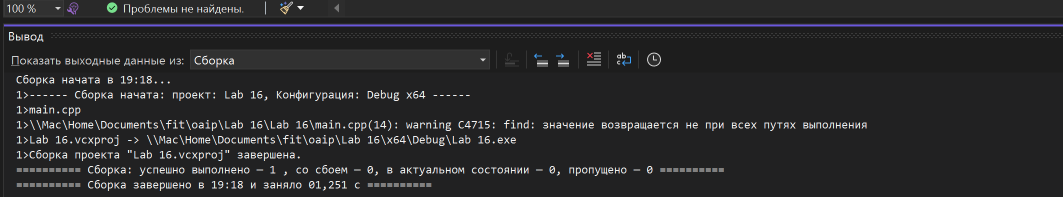
    }

    cout << "Строка, которая содержит d: " << find(a, d);

**delete**[] a; //очищаем память, выделенную под массив

}





2. В строке подсчитать сумму кодов символов каждого слова. Слова в строке разделены пробелами.

#include <iostream> //для работы с вводом/выводом

#include <string> //для работы со строками

**using** **namespace** std; //объявляем программе, что использем пространство имен std

//функция для нахождения суммы кодов

**int** finds(string s) {

**int** sum = 0;

**for** (**int** i = 0; i < s.length(); i++) {

**if** (s[i] != ' ') {

            sum += **int**(s[i]);

        }

    }

**return** sum;

}

**int** main() {

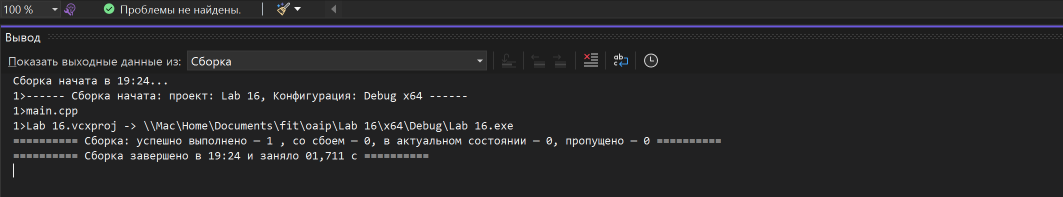
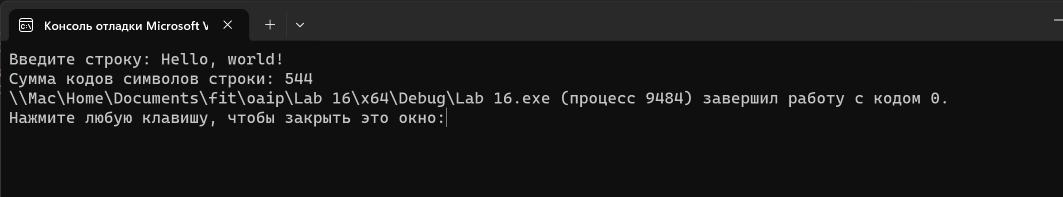
    setlocale(LC\_ALL, "rus"); //добовляем возможность работы с русским языком

    string s;//создание строки

    cout << "Введите строку: "; cin >> s;

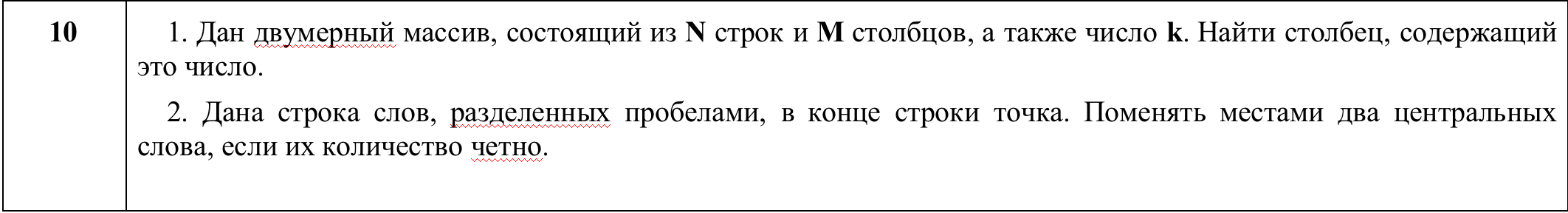
    cout << "Сумма кодов символов строки: " << finds(s);

}



Вариант 10

4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.



1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов, а также число **k**. Найти столбец, содержащий это число.

#include <iostream> //для работы с вводом/выводом

#include <string> //для работы со строками

**using** **namespace** std; //объявляем программе, что использем пространство имен std

//функция для поиска символа в столбце

**int** findi(**int**\*\* a, **int** d) {

**for** (**int** i = 0; i < **sizeof**(a[0]); i++) {

**for** (**int** j = 0; j < **sizeof**(a[0][0]); j++) {

**if** (a[i][j] == d) {

**return** j + 1;

            }

        }

    }

}

**int** main() {

    setlocale(LC\_ALL, "rus"); //добавляем возможность работы с русским языком

**int** n, m, k;

    cout << "Введите размеры матрицы: "; cin >> n >> m;

    cout << "Введите k: "; cin >> k;

    cout << "Введите массив: \n";

    //создание массива и выделения памяти

**int**\*\* a;

    a = **new** **int**\* [n];

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

        a[i] = **new** **int**[m];

    }

    //ввод элементов в массив

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

**for** (**int** j = 0; j < m; j++) {

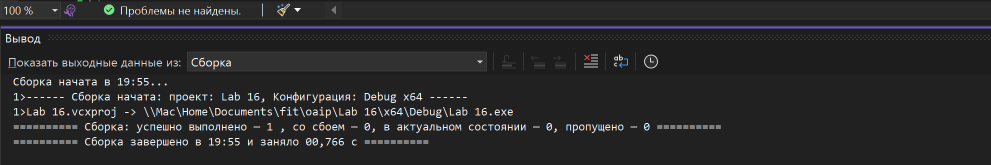
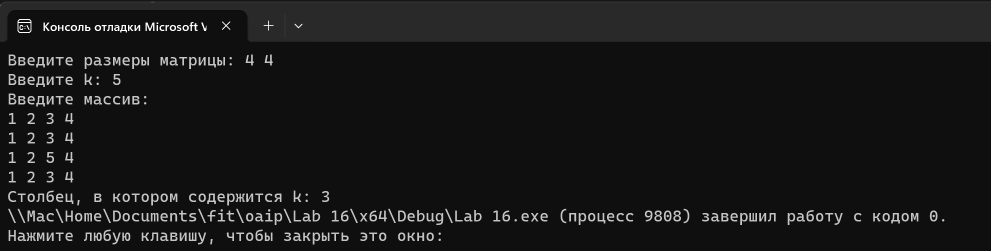
            cin >> a[i][j];

        }

    }

    cout << "Столбец, в котором содержится k: " << findi(a, k);

}



2. Дана строка слов, разделенных пробелами, в конце строки точка. Поменять местами два центральных слова, если их количество четно.

#include <iostream> // Для работы с вводом/выводом

#include <string>    // Для работы со строками

#include <vector>    // Для использования векторов

**using** **namespace** std; //объявляем программе, что используем пространство имен std

**int** main() {

    setlocale(LC\_ALL, "rus"); //добовляем возможность работы с русским языком

    vector<string> a; // Векторный массив строк для хранения отдельных слов

    cout << "Введите предложение: ";

    string word, s;

**char** let;

    // Ввод исходной строки с остановкой на точке

    getline(cin, s, '.');

    s += "."; // Добавление точки в конец строки для обработки последнего слова

**int** i = 0;

    let = s[i];

    // Цикл разбора строки на отдельные слова

**while** (let != '.') {

**if** (let != ' ') {

            word += let; // Отделение отдельных слов

        } **else** {

            a.push\_back(word); // Заполнение массива строк

            word = "";

        }

        i++;

        let = s[i];

    }

    a.push\_back(word); // Добавление последнего слова в массив

    word = "";

    // Если количество слов четное, меняем местами слова в середине

**if** (a.size() % 2 == 0) {

        swap(a[a.size() / 2], a[a.size() / 2 - 1]);

    }

    cout << "Новое предложение: ";

    // Вывод слов в обновленном порядке

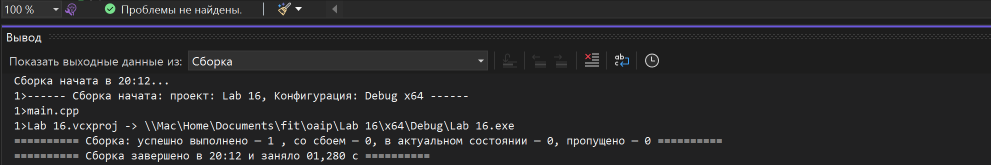
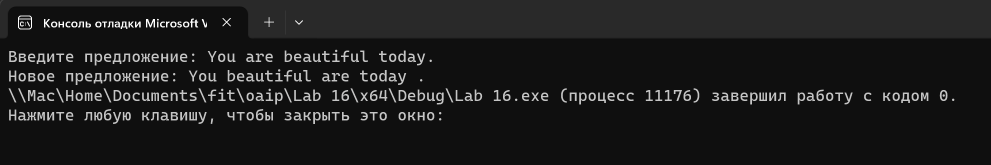
**for** (**auto** c = a.begin(); c != a.end(); c++) {

        cout << \*c << " ";

    }

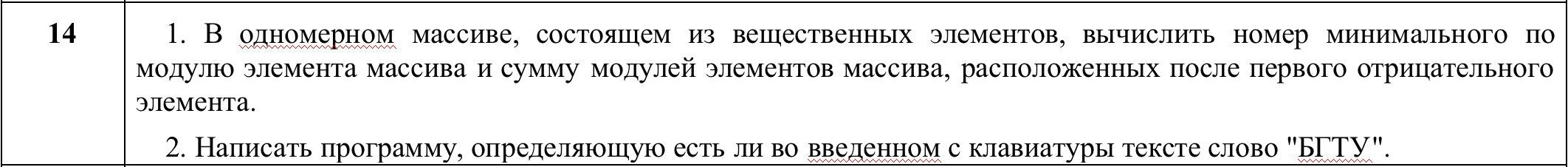
    cout << ".";

}



Вариант 14

4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.



1. В одномерном массиве, состоящем из вещественных элементов, вычислить номер минимального по модулю элемента массива и сумму модулей элементов массива, расположенных после первого отрицательного элемента.

#include <iostream> // Для работы с вводом/выводом

**using** **namespace** std; //объявляем программе, что используем пространство имен std

//функция для поиска минимального элемента

**float** find\_min(**float**\* arr, **int** n) {

**int** a = 0;

**for** (**int** i = 1; i < n; i++) {

**if** (abs(arr[a]) > abs(arr[i])) {

            a = i;

        }

    }

**return** arr[a];

}

//функция для нахождения суммы

**float** find\_sum(**float**\* arr, **int** n) {

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

**if** (arr[i] < 0) { //поиск первого отрицательного

**float** sum = 0;

**for** (**int** j = i + 1; j < n; j++) { //считаю сумму

                sum += arr[j];

            }

**return** sum;

        }

    }

}

**int** main() {

    setlocale(LC\_ALL, "rus"); //добавляем возможность работы с русским языком

**float** n;

**float**\* arr; //создаем массив

    cout << "Введите размер массива: "; cin >> n;

    arr = **new** **float**[n]; //задаем размер массива

    cout << "Введите массив: ";

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

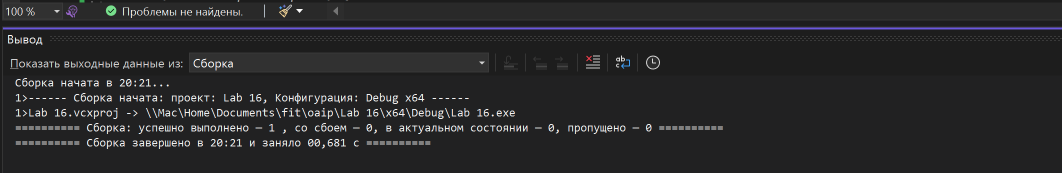
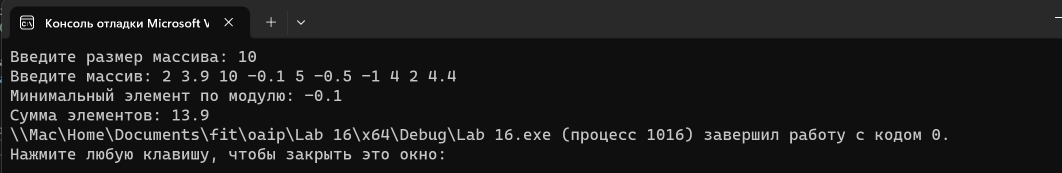
        cin >> arr[i];

    }

    cout << "Минимальный элемент по модулю: " << find\_min(arr, n) << endl;

    cout << "Сумма элементов: " << find\_sum(arr, n); //вывод

}



2. Написать программу, определяющую есть ли во введенном с клавиатуры тексте слово "БГТУ".

#include <iostream> //для работы с вводом/выводом

#include <string> //для работы со строками

#include <Windows.h> //для работы с опирационной системой windows

**using** **namespace** std; //объявляем программе, что используем пространство имен std

//функция для нахождения слова БГТУ

string find\_bstu(string s) {

**for** (**int** i = 0; i < s.length(); i++) {

**if** (s[i] == 'Б' && s[i + 1] == 'Г' && s[i + 2] == 'Т' && s[i + 3] == 'У') {

**return** "ДА";

        }

    }

**return** "НЕТ";

}

**int** main() {

    SetConsoleCP(1251); //установка кодировки

    setlocale(LC\_ALL, "rus"); //добавляем возможность работы с русским языком

    string s; //создаем строку

    cout << "Введите строку: "; getline(cin, s);

    cout << "В строке содержится БГТУ: " << find\_bstu(s);

}