**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

Лабораторная работа 16

По дисциплине «Основы алгоритмизации и программирования»

На тему «Функции пользователя»

Выполнила:

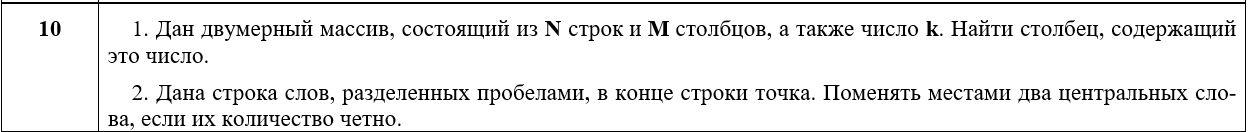
Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск

Дополнительные задания к лаб 16



#include <iostream>

using namespace std;

int find\_k(int\*\* x, int s) //объявляет функцию find\_k, которая принимает двумерный массив целых чисел x и целое число s в качестве аргументов

{

for (int i = 0; i < sizeof(x[0]); i++)//цикл перебирает индексы строк массива x. sizeof(x[0]) возвращает размер строки в байтах

for (int j = 0; j < sizeof(x[0][0]); j++) //вложенный цикл перебирает индексы столбцов в каждой строке массива x.

{

if (x[i][j] == s)

return j + 1;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

int y, m, z;//объявляет переменные y, m и z для хранения размеров матрицы и значения k.

cout << "Размер матрицы: "; cin >> y >> m;

cout << "k: "; cin >> z;

cout << "Массив: \n";

int\*\* x;

x = new int\* [y];//выделяет память под двумерный массив x размером y на m.

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

x[i] = new int[m];

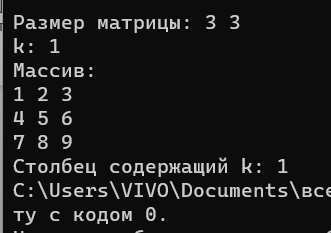
for (int i = 0; i < y; i++)//Циклы для заполнения матрицы из пользовательского ввода

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

cin >> x[i][j];

cout << "Столбец содержащий k: " << find\_k(x, z);//вызывает функцию find\_k для поиска значения k в матрице и выводит индекс столбца, содержащего это значение

}



#include <iostream>

#include <list>

#include <string>

#include <vector>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

vector<string> t; //объявляет вектор t для хранения слов из предложения.

cout << "Начальное предложение: ";

string sl;//объявляет строку sl для хранения текущего слова.

string x;//объявляет строку x для хранения всего предложения.

char bol;//объявляет символьную переменную bol для хранения текущего символа.

getline(cin, x, '.'); // считывает предложение до точки и сохраняет его в строку x.

x += ".";//добавляет точку в конец строки x.

int i = 0;

bol = x[i];

while (bol != '.')//начинает цикл, который выполняется до тех пор, пока значение bol не равно точке.

{

if (bol != ' ')//если текущий символ не пробел, добавляет его к текущему слову (sl), иначе добавляет текущее слово в вектор t, очищает переменную sl.

{

sl += bol;

}

else

{

t.push\_back(sl);

sl = "";

}

i++;

bol = x[i];

}

t.push\_back(sl);//добавляет последнее слово в вектор t и очищает переменную sl.

sl = "";

if (size(t) % 2 == 0)//если количество слов в предложении четное, меняет местами два средних слова

{

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

}

cout << "Конечное предложение: ";//Цикл для вывода каждого слова из вектора t.

for (auto c = t.begin(); c != t.end(); c++)

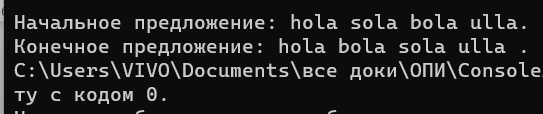
{

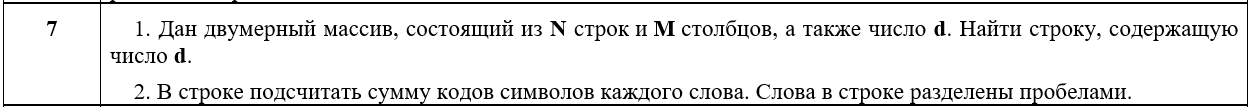
cout << \*c << " ";

}

cout << ".";

}





#include <iostream>

using namespace std;

int find\_d(int\*\* x, int d) //объявляет функцию find\_d, которая принимает двумерный массив целых чисел x и целое число d в качестве аргументов

{

for (int i = 0; i < sizeof(x[0]); i++)//начинает цикл, который перебирает элементы первого измерения массива x. sizeof(x[0]) возвращает размер первого измерения массива x.

for (int j = 0; j < sizeof(x[0][0]); j++) //начинает вложенный цикл, который перебирает элементы второго измерения массива x

{

if (x[i][j] == d)//если текущий элемент массива равен d, функция возвращает индекс строки (увеличенный на 1, так как обычно индексы начинаются с 0).

return i + 1;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

int p, q, d;

cout << "Размер массива: "; cin >> q >> p;

cout << "d: "; cin >> d;

cout << "Массив: \n";

int\*\* x;

x = new int\* [q];//выделяет память под двумерный массив x

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

x[i] = new int[p];

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

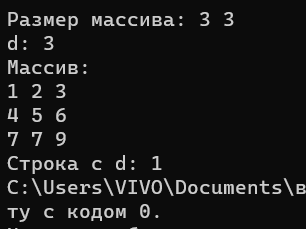
for (int j = 0; j < p; j++)

cin >> x[i][j];

cout << "Строка с d: " << find\_d(x, d);// выводит результат вызова функции find\_d с массивом x и значением d.

delete[] x;// освобождает память, выделенную под массив x.

}



#include <iostream>

#include <string>

using namespace std;

int finds(string с) //объявляет функцию finds, которая принимает строку с в качестве аргумента

{

int sum = 0; // объявляет переменную sum и инициализирует её значением 0. Переменная sum будет использоваться для накопления суммы кодов символов.

for (int i = 0; i < с.length(); i++)//начинает цикл, который перебирает все символы в строке с.

{

if (с[i] != ' ')//проверяет, не является ли текущий символ пробелом.

sum += int(с[i]); // добавляет код символа к сумме. Функция int(с[i]) приводит символ к его числовому представлению (коду ASCII) и прибавляет его к переменной sum.

}

return sum;

}

int main()

{

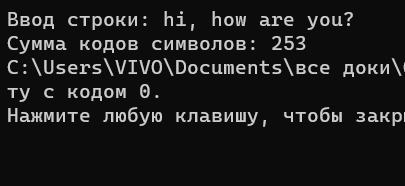
setlocale(LC\_ALL, "rus");

string с;// объявляет переменную с типа string, которая будет использоваться для хранения введенной строки.

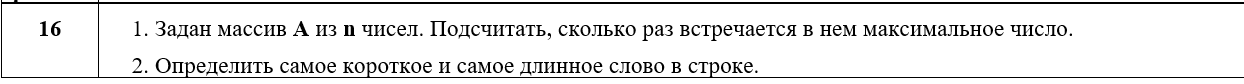
cout << "Ввод строки: "; cin >> с;

cout << "Сумма кодов символов: " << finds(с);

}



Вариант 16



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int n;

cout << "Размер массива: ";

cin >> n;

int\* arr = new int[n];//Создает динамический массив целых чисел размером n и сохраняет указатель на его начало в переменной arr.

cout << "Элементы массива:" << endl;

for (int i = 0; i < n; i++) {//Считывает значения элементов массива из ввода пользователя и сохраняет их в соответствующие ячейки массива arr.

cin >> arr[i];

}

int maxNum = arr[0];

int count = 1;

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

if (arr[i] > maxNum) {//сли текущий элемент больше maxNum, обновляет maxNum и сбрасывает счетчик count на 1.

maxNum = arr[i];

count = 1;

}

else if (arr[i] == maxNum) {//Если текущий элемент равен maxNum, увеличивает счетчик count.

count++;

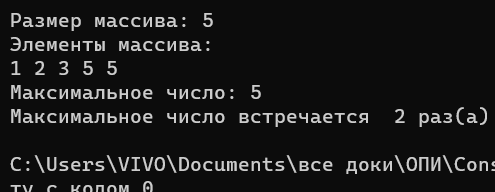
}

}

cout << "Максимальное число: " << maxNum << endl;

cout << "Максимальное число встречается " << count <<" раз(а)" << endl;

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char words[100];//объявляет массив символов words размером 100, который будет использоваться для хранения введенной строки.

cout << "Предложение: ";

cin.getline(words, 100);//Считывает строку из ввода пользователя и сохраняет ее в массив words.

char shorte[100], longe[100];//Объявляет массивы символов shorte и longe, которые будут использоваться для хранения самого короткого и самого длинного слов соответственно.

int StartIndex = -1;//Инициализирует переменную StartIndex значением -1, которая будет использоваться для отслеживания начала каждого слова.

//Инициализируют переменные shorteL и longeL значением -1, которые будут использоваться для отслеживания длины самого короткого и самого длинного слов соответственно.

int shorteL = -1;

int longeL = -1;

int index = 0;//Инициализирует переменную index значением 0, которая будет использоваться для итерации по символам в строке words.

while (words[index] != '\0') {//Цикл проходит по каждому символу в строке words до тех пор, пока не достигнет нулевого символа, обозначающего конец строки.

// Внутри цикла происходит разбор строки на отдельные слова и определение самого короткого и самого длинного слова.

while (words[index] == ' ') {

index++;

}

if (words[index] != '\0' && words[index] != ' ') {//Проверяет, что текущий символ не является нулевым символом (\0) и не является пробелом. Если это так, то устанавливает StartIndex равным текущему индексу в строке words.

StartIndex = index;

}

while (words[index] != '\0' && words[index] != ' ') {//цикл проверяет каждый символ до тех пор, пока не достигнет нулевого символа или пробела. В этом процессе index увеличивается на 1 для перехода к следующему символу.

index++;

}

int wordLength = index - StartIndex;//вычисляется длина слова путем вычитания StartIndex из index.

if (shorteL == -1 || wordLength < shorteL) {//проверяется, является ли текущее слово самым коротким или самым длинным. Если переменная shorteL равна -1 (т.е. это первое слово), или длина текущего слова меньше shorteL, то обновляется переменная shorteL и копируется соответствующее слово в массив shorte. Аналогично для самого длинного слова.

shorteL = wordLength;

strncpy\_s(shorte, words + StartIndex, shorteL);

shorte[shorteL] = '\0';

}

if (longeL == -1 || wordLength > longeL) {

longeL = wordLength;

strncpy\_s(longe, words + StartIndex, longeL);

longe[longeL] = '\0';

}

}

cout << "Самое короткое слово: " << shorte << endl;

cout << "Самое длинное слово: " << longe << endl;

}

