Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа №16

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Функцыі карыстальніка»

Выканала:

Студэнтка 1 курса 6 группы

Літвінчук Дар'я Валер'еўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

**Варыянт 10**

|  |  |
| --- | --- |
| **10** | 1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов, а также число **k**. Найти столбец, содержащий это число.  2. Дана строка слов, разделенных пробелами, в конце строки точка. Поменять местами два центральных слова, если их количество четно. |

using namespace std;

void findColumnWithNumber(int\*\* arr, int N, int M, int k) {

bool found = false;

// Перебираем каждый столбец

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

// Проверяем, содержится ли число k в текущем столбце

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

if (arr[i][j] == k) {

found = true;

cout << "Столбец с числом " << k << " найден. Номер столбца: " << j + 1 << endl;

break;

}

}

if (found) {

break; // Если число найдено в столбце, прекращаем поиск

}

}

if (!found) {

cout << "Число " << k << " не найдено в массиве.\n";

}

}

int main() {

setlocale(LC\_ALL, "RU");

int N, M, k;

cout << "Введите количество строк N: ";

cin >> N;

cout << "Введите количество столбцов M: ";

cin >> M;

// Выделяем память под двумерный массив

int\*\* arr = new int\* [N];

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

arr[i] = new int[M];

}

cout << "Введите элементы массива:\n";

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

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

cin >> arr[i][j];

}

}

cout << "Введите число k: ";

cin >> k;

findColumnWithNumber(arr, N, M, k);

// Освобождаем выделенную память

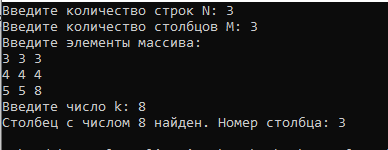
for (int i = 0; i < N; ++i) {

delete[] arr[i];

}

delete[] arr;

}



#include <iostream>

#include <string>

#include <vector>

#include <Windows.h>

using namespace std;

void swapCentralWords(string& sentence) {

vector<string> words;

string word;

// Разбиваем строку на слова

for (char ch : sentence) {

if (ch != ' ' && ch != '.') {

word += ch;

}

else {

words.push\_back(word);

word.clear();

}

}

int wordCount = words.size();

// Проверяем, четное ли количество слов

if (wordCount % 2 == 0 && wordCount >= 4) {

// Меняем местами два центральных слова

int middle1 = (wordCount - 1) / 2;

int middle2 = wordCount / 2;

swap(words[middle1], words[middle2]);

// Формируем новую строку

sentence.clear();

for (const string& w : words) {

sentence += w + ' ';

}

sentence.pop\_back(); // Убираем последний пробел

sentence += '.';

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string sentence;

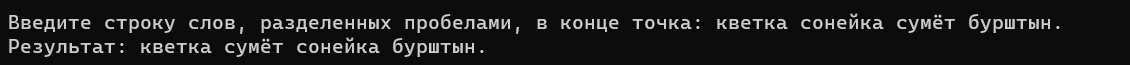
cout << "Введите строку слов, разделенных пробелами, в конце точка: ";

getline(cin, sentence);

swapCentralWords(sentence);

cout << "Результат: " << sentence << endl;

}



**дадатковыя заданні:**

|  |  |
| --- | --- |
| **16** | 1. Задан массив **A** из **n** чисел. Подсчитать, сколько раз встречается в нем максимальное число.  2. Определить самое короткое и самое длинное слово в строке. |

using namespace std;

int countMaxOccurrences(const vector<int>& arr) {

if (arr.empty()) {

return 0;

}

int maxNumber = arr[0];

int count = 1;

for (size\_t i = 1; i < arr.size(); ++i) {

if (arr[i] > maxNumber) {

maxNumber = arr[i];

count = 1;

}

else if (arr[i] == maxNumber) {

count++;

}

}

return count;

}

int main() {

setlocale(LC\_ALL, "RU");

int n;

cout << "Введите размер массива: ";

cin >> n;

vector<int> arr(n);

cout << "Введите элементы массива:\n";

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

cin >> arr[i];

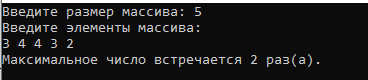
}

int occurrences = countMaxOccurrences(arr);

cout << "Максимальное число встречается " << occurrences << " раз(а).\n";

return 0;

}



#include <iostream>

#include <sstream>

#include <vector>

using namespace std;

pair<string, string> findShortestAndLongestWord(const string& sentence) {

istringstream iss(sentence);

string word;

string shortestWord, longestWord;

while (iss >> word) {

// Инициализируем самое короткое и самое длинное слово, если это первое слово

if (shortestWord.empty()) {

shortestWord = word;

}

if (longestWord.empty()) {

longestWord = word;

}

// Сравниваем текущее слово с самым коротким и самым длинным

if (word.length() < shortestWord.length()) {

shortestWord = word;

}

if (word.length() > longestWord.length()) {

longestWord = word;

}

}

return make\_pair(shortestWord, longestWord);

}

int main() {

setlocale(LC\_ALL, "RU");

string sentence;

cout << "Введите строку: ";

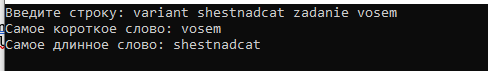
getline(cin, sentence);

pair<string, string> result = findShortestAndLongestWord(sentence);

cout << "Самое короткое слово: " << result.first << endl;

cout << "Самое длинное слово: " << result.second << endl;

}



|  |  |
| --- | --- |
| **8** | 1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов. Найти количество отрицательных, положительных и нулевых элементов массива.  2. Написать программу, которая вводит несколько слов текста и определяет общее количество слов. Слова разделены пробелами. |

#include <iostream>

using namespace std;

void countElements(int\*\* arr, int N, int M, int& negCount, int& posCount, int& zeroCount) {

negCount = 0;

posCount = 0;

zeroCount = 0;

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

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

if (arr[i][j] < 0) {

negCount++;

}

else if (arr[i][j] > 0) {

posCount++;

}

else {

zeroCount++;

}

}

}

}

int main() {

setlocale(LC\_ALL, "RU");

int N, M;

cout << "Введите количество строк N: ";

cin >> N;

cout << "Введите количество столбцов M: ";

cin >> M;

// Выделяем память под двумерный массив

int\*\* arr = new int\* [N];

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

arr[i] = new int[M];

}

cout << "Введите элементы массива:\n";

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

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

cin >> arr[i][j];

}

}

int negCount, posCount, zeroCount;

countElements(arr, N, M, negCount, posCount, zeroCount);

cout << "Количество отрицательных элементов: " << negCount << endl;

cout << "Количество положительных элементов: " << posCount << endl;

cout << "Количество нулевых элементов: " << zeroCount << endl;

// Освобождаем выделенную память

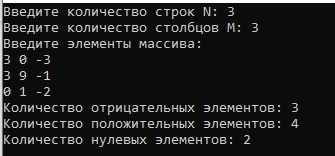
for (int i = 0; i < N; ++i) {

delete[] arr[i];

}

delete[] arr;

}

  
  
#include <iostream>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL, "RU");

string text;

cout << "Введите текст: ";

getline(cin, text);

// Инициализируем счетчик слов

int wordCount = 0;

// Переменная, чтобы определить, начинается ли новое слово

bool newWord = true;

// Проходим по каждому символу в тексте

for (char ch : text) {

// Если текущий символ - пробел, устанавливаем флаг нового слова

if (ch == ' ' || ch == '\t' || ch == '\n') {

newWord = true;

}

else {

// Если новое слово, увеличиваем счетчик слов

if (newWord) {

wordCount++;

newWord = false; // Сбрасываем флаг нового слова

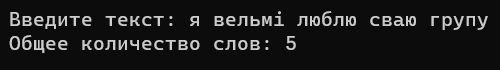
}

}

}

cout << "Общее количество слов: " << wordCount << endl;

}



|  |  |
| --- | --- |
| **14** | 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.  2. Написать программу, определяющую есть ли во введенном с клавиатуры тексте слово "БГТУ". |

#include <iostream>

#include <cmath>

using namespace std;

// Функция для вычисления номера минимального по модулю элемента массива

int findMinModElementIndex(float arr[], int size) {

int minModIndex = 0;

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

if (abs(arr[i]) < abs(arr[minModIndex])) {

minModIndex = i;

}

}

return minModIndex;

}

// Функция для вычисления суммы модулей элементов после первого отрицательного элемента

float sumModAfterNegative(float arr[], int size) {

int firstNegativeIndex = -1;

// Находим индекс первого отрицательного элемента

for (int i = 0; i < size; ++i) {

if (arr[i] < 0) {

firstNegativeIndex = i;

break;

}

}

// Если нет отрицательных элементов, возвращаем 0

if (firstNegativeIndex == -1) {

return 0.0;

}

// Считаем сумму модулей элементов после первого отрицательного

float sum = 0.0;

for (int i = firstNegativeIndex + 1; i < size; ++i) {

sum += abs(arr[i]);

}

return sum;

}

int main() {

// Устанавливаем русскую локаль

setlocale(LC\_ALL, "Russian");

int size;

cout << "Введите размер массива: ";

cin >> size;

// Выделяем динамическую память под массив

float\* arr = new float[size];

// Вводим элементы массива

cout << "Введите элементы массива:\n";

for (int i = 0; i < size; ++i) {

cin >> arr[i];

}

int choice;

cout << "Выберите операцию:\n";

cout << "1. Номер минимального по модулю элемента\n";

cout << "2. Сумма модулей элементов после первого отрицательного\n";

cin >> choice;

switch (choice) {

case 1: {

// Вычисляем номер минимального по модулю элемента

int minModIndex = findMinModElementIndex(arr, size);

cout << "Номер минимального по модулю элемента: " << minModIndex << endl;

break;

}

case 2: {

// Вычисляем сумму модулей элементов после первого отрицательного

float sumAfterNegative = sumModAfterNegative(arr, size);

cout << "Сумма модулей элементов после первого отрицательного: " << sumAfterNegative << endl;

break;

}

default:

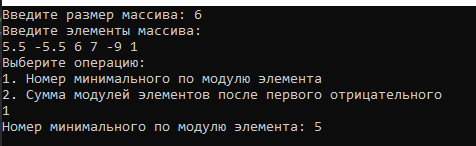
cout << "Некорректный выбор операции.\n";

}

// Освобождаем выделенную память

delete[] arr;

}



#include <iostream>

#include <string>

using namespace std;

// Функция для определения наличия слова "БГТУ" в тексте

bool containsBGTU(const string& text) {

size\_t found = text.find("BSTU");

return found != string::npos;

}

int main() {

// Устанавливаем русскую локаль

setlocale(LC\_ALL, "Russian");

while (true) {

cout << "Введите текст (0 для выхода): ";

string text;

getline(cin, text);

if (text == "0") {

break;

}

if (containsBGTU(text)) {

cout << "Слово 'БГТУ' найдено в тексте.\n";

}

else {

cout << "Слово 'БГТУ' не найдено в тексте.\n";

}

}

}

