|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта (ИИИ)  Кафедра промышленной информатики (ПИ) |

**ОТЧЁТ ПО**

**ПРАКТИЧЕСКОЙ РАБОТЕ №5**

по дисциплине «Информатика»

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы | Давлетов Э.А., ИВБО–21–24 |  |  |
|  | *(Ф.И.О., учебная группа)* |  | *(подпись студента)* |
| Руководитель лаб. работы |  |  |  |
|  | *(Ф.И.О.)* |  | *(подпись руководителя)* |
| Лабораторная работа представлена к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024г. |  |  |
|  |  |  |  |

Москва, 2024г.

Задача 1 & 2 (Евклид, Эратосфен)

#include <iostream>

using namespace std;

int NODmod(int a, int b) {

int t;

while (b != 0) {

t = b;

b = a % b;

a = t;

}

return a;

}

int NODminus(int a, int b) {

int t;

while (a!=b) {

if (a > b)

a -= b;

else

b -= a;

}

return a;

}

int simples(int number) {

int counter = 0;

for (int i = 2; i <= number; i++){

bool flag = true;

for (int j = 2; j <= i/2; j++) {

if (i % j == 0) {

flag = false;

}

}

if (flag == true){

counter += 1;

cout << i << " ";

}

}

cout << "count: ";

return counter;

}

int main() {

int m1, m2;

cout << "enter first num: ";

cin >> m1;

cout << "enter second num: ";

cin >> m2;

cout << NODmod(m1, m2) << endl;

cout << NODminus(m1, m2) << endl;

int pr;

cout << "enter num: ";

cin >> pr;

cout << simples(pr);

}

Задача 3 (Обработка 16)

#include <iostream>

#include <cstdlib>

#include <fstream>

#include <windows.h>

#include <string>

using namespace std;

// 16. Поиск определенного сочетания слов в текстовом файле

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char buff[40];

ifstream fin("D:\\somedocuments\\informatika\\test6.txt");

if (!fin.is\_open()) {

cout << "file wasn't open, try again" << endl;

}

else {

string needfind;

cout << "What do you want to find? Enter: ";

getline(cin, needfind); // getline для считывания строки

int line = 0;

while (!fin.eof()) {

fin.getline(buff, 40);

string text = buff;

line++;

// Проверка на наличие подстроки в строке

size\_t pos = text.find(needfind);

if (pos != string::npos) {

cout << "Found at pos = " << pos << ", line = " << line << ", words - '" << needfind << "' " << endl;

}

}

fin.close();

}

return 0;

}

Задача 4 (Обработка 28)

#include <iostream>

#include <cstdlib>

#include <fstream>

#include <windows.h>

#include <string>

using namespace std;

// 28. Преобразование текста в массив символов.

int main() {

SetConsoleCP(1251); SetConsoleOutputCP(1251);

char buff[40];

ifstream fin("D:\\somedocuments\\informatika\\test6.txt");

if (!fin.is\_open()) {

cout << "file wasn't open, try again" << endl;

}

else {

while (!fin.eof()) {

fin.getline(buff, 40);

cout << buff << " === type - ";

cout << typeid(buff).name() << endl;

}

fin.close();

}

return 0;

}

Задача 5 (Ряды 1)

#include <iostream>

#include <cmath>

using namespace std;

// 1. Дано целое число n (вводится с клавиатуры). Вычислить:

int main()

{

int n;

float sum = 0;

float b = 0;

cout << "Enter number n = ";

cin >> n;

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

b += sin(i);

sum += i / b;

}

cout << sum;

}

Задача 6 (Ряды 65)

#include <iostream>

#include <cmath>

using namespace std;

// 65. Задан числовой массив F[1:n]. Найти отрезок массива максимальной длины, в котором первое число равно

// последнему, второе - предпоследнему и т.д. Напечатать длину этого отрезка.

int N;

cout << "enter array size: ";

cin >> N;

int\* a = new int[N];

cout << "Enter numbers to array" << endl;

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

cin >> a[i]; // вводим числа в массив

}

int max = 1;

for (int i = 0; i < N - 1; ++i) // проходим по массивуу

if (a[i] == a[i + 1]) { // если четный (3, 4, 4, 3)

int j = 1;

while ((i - j > -1) and (i + j + 1 < N) and (a[i - j] == a[i + j + 1])) // проверка на расположение одинаковых цифр и то что всё норм

++j; // инкрементируем

if (max < 2 \* j) // проверка на максимум

max = 2 \* j; // присваиваем помноженную на 2 (у нас четный отрезок)

}

else if (i > 0 and (a[i - 1] == a[i + 1])) { // если нечетный (3, 4, 5, 4, 3)

int j = 1;

while (i + j < N and a[i - j] == a[i + j]) // пока элементы через один равны

++j; // инкрементируем

if (max < 2 \* j + 1)

max = 2 \* j - 1; // -1 так как нечетный отрезок

}

cout << max;

delete[] a;

}

Задача 7 (Файл 24)

#include <iostream>

#include <iomanip>

#include <string>

#include <fstream>

using namespace std;

const int NUM\_COUNTRIES = 10; //колво стран

void calculateResults(int gold[], int silver[], int bronze[], int total[], int points[], int n) { // расчёт медалей и очков стран

for (int i = 0; i < n; i++) { // проходимся по каждой стране

total[i] = gold[i] + silver[i] + bronze[i]; // сумма медалей всех

points[i] = gold[i] \* 7 + silver[i] \* 6 + bronze[i] \* 5; // сумма очков всех

}

}

void sortAlphabetically(string countries[], int gold[], int silver[], int bronze[], int total[], int points[], int n) {

for (int i = 0; i < n - 1; i++) { // сортировка по алфавиту пузырьковая внеш цикл

for (int j = 0; j < n - i - 1; j++) { // для соседних элементов, внутр цикл

if (countries[j] > countries[j + 1]) {

// свап стран

swap(countries[j], countries[j + 1]);

// свап медалей и очков для того чтобы нормально было всё

swap(gold[j], gold[j + 1]);

swap(silver[j], silver[j + 1]);

swap(bronze[j], bronze[j + 1]);

swap(total[j], total[j + 1]);

swap(points[j], points[j + 1]);

}

}

}

}

void saveResultsToFile(const string countries[], int gold[], int silver[], int bronze[], int total[], int points[], int n, const string& filename) {

ofstream file(filename); // функция для сохранения в файл, открываем файл

if (file.is\_open()) { // если открыт

file << "Итоги Олимпийских игр:\n";

file << "Страна Золото Серебро Бронза Всего Очки\n"; // вписываем в файлик

for (int i = 0; i < n; i++) { // вписываем циклом в файл столбцы и строки

file << setw(10) << countries[i] << setw(8) << gold[i]

<< setw(9) << silver[i] << setw(8) << bronze[i]

<< setw(7) << total[i] << setw(6) << points[i] << "\n";

}

file.close(); // закрываем

cout << "Результаты сохранены в файл: " << filename << endl; // всё норм сохранилось

}

else { // если не открылся

cerr << "Ошибка: не удалось открыть файл для записи.\n";

}

}

void printResults(const string countries[], int gold[], int silver[], int bronze[], int total[], int points[], int n) {

cout << "Итоги Олимпийских игр:\n";

cout << "Страна Золото Серебро Бронза Всего Очки\n"; // это собственно принтим

for (int i = 0; i < n; i++) { // циклом выводим данные стран по всему что есть

cout << setw(10) << countries[i] << setw(8) << gold[i] // setw для ширины чтобы красиво

<< setw(9) << silver[i] << setw(8) << bronze[i]

<< setw(7) << total[i] << setw(6) << points[i] << "\n";

}

}

int main() {

setlocale(LC\_ALL, "rus");

string countries[NUM\_COUNTRIES] = { "Германия", "Норвегия", "Россия", "Австрия", "Канада", "США", "Финляндия", "Япония", "Китай", "Корея" };

int gold[NUM\_COUNTRIES] = { 12, 10, 9, 3, 6, 6, 2, 5, 0, 3 }; // сверху массив строк для каждой страны

int silver[NUM\_COUNTRIES] = { 9, 10, 6, 5, 5, 3, 4, 1, 6, 1 }; // массив серебр медалей

int bronze[NUM\_COUNTRIES] = { 8, 5, 3, 9, 4, 4, 6, 4, 2, 2 }; // массив бронзовых

int total[NUM\_COUNTRIES]; // тут колво очков и медалей в двух массивах

int points[NUM\_COUNTRIES];

calculateResults(gold, silver, bronze, total, points, NUM\_COUNTRIES); // рассчитываем резы

sortAlphabetically(countries, gold, silver, bronze, total, points, NUM\_COUNTRIES); // сортируем

printResults(countries, gold, silver, bronze, total, points, NUM\_COUNTRIES); // принтим

saveResultsToFile(countries, gold, silver, bronze, total, points, NUM\_COUNTRIES, "results.txt"); // записываем в файлик

return 0;

}

Задача 8 (Шарики)

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int result = 0; // счетчик итоговый

void generate(vector<int>& a, int t, int n) { // для перестановок начиная с индекса t, рекурсия

if (t == n) { // проверка что все перестановки использовали

bool found = false; // типа флаг на совпадение нужное нам

for (int i = 0; i < n; ++i) { // проходимся по шарикам в перестановке

if (a[i] == i + 1) { // шарик a[i] стоит на своем месте (i+1 тк индексы с нуля)

found = true;

break; // меняем и выходим из цикла

}

}

if (found) { // если в флаг true на итерации

result++; // то увеличиваем счётчик

}

return;

}

for (int j = t; j < n; ++j) { // перестановки от t до текущего элемента

swap(a[t], a[j]); // меняем местами

generate(a, t + 1, n); // вызываем следующую позицию в перестановке

swap(a[t], a[j]); // массив в исходное состояние

}

}

int main() {

int n;

cout << "number of balls: ";

cin >> n;

if (n <= 0) {

cout << "wrong, must number of balls must be >0" << endl;

return 1;

}

vector<int> a(n); // Массив для хранения перестановки

// Заполняем массив числами от 1 до n

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

a[i] = i + 1;

}

// генерируем все перестановки и считаем количество совпадений

generate(a, 0, n);

cout << "number of good situations: " << result << endl;

return 0;

}

Задача 10 (Структурные числа)

#include <iostream>

#include <cmath>

using namespace std;

bool primes(int number) {

if (number <= 1) return false;

if (number == 2) return true;

if (number % 2 == 0) return false;

for (int i = 3; i <= sqrt(number); i += 2) {

if (number % i == 0) {

return false;

}

}

return true;

}

int main() {

int m, n;

cout << "Enter the range [m, n]: ";

cin >> m >> n;

cout << "numbers from m to n are: ";

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

if (primes(p) and primes(2 \* p + 1)) {

cout << p << " ";

}

}

cout << endl;

return 0;

}