7 лаба

#include <Windows.h>

#include <iostream>

using namespace std;

void matrix69();

void matrix27();

const int M = 20, N = 20; //максимальне кількість рядків стовпців

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int menu;

do

{

cout << "Номер завдання(0 для виходу):";

cin >> menu;

switch (menu)

{ // перемикання між завданнями

case 1:

matrix27();

break; // Завдання 1

case 2:

matrix69();

break; // Завдання 2

}

} while (menu);

system("pause");

return 0;

}

void input\_matrix(int matrix[M][N], int rows, int cols)

{

cout << "Введіть матрицю розміру " << rows << " x " << cols << ":" << endl;

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

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

cin >> matrix[i][j];

}

}

}

// функція виведення масиву

void print\_matrix(int matrix[M][N], int rows, int cols) {

cout << "Матриця розміру " << rows << " x " << cols << ":" << endl;

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void matrix27()

{

int matrix[M][N];

int cols, rows, min, min\_max = 0;;

cout << "Введіть кількість рядків (максимум " << M << "): ";

cin >> rows;

cout << "Введіть кількість стовпців (максимум " << N << "): ";

cin >> cols;

input\_matrix(matrix, rows, cols);

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

{

int min = matrix[rows][0];

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

{

if (j == 0 || matrix[i][j] < min)

{

min = matrix[i][j];

}

if (i == 0 || min > min\_max)

{

min\_max = min;

}

}

}

cout << "Min max" << min\_max << endl;

}

void matrix69()

{

int rows, cols, k;

int matrix[M][N];

// Введення розмірів матриці

cout << "Введіть кількість рядків (максимум " << M << "): ";

cin >> rows;

cout << "Введіть кількість стовпців (максимум " << N << "): ";

cin >> cols;

cout << "K:";

cin >> k;

input\_matrix(matrix, rows, cols);

// Введення та виведення матриці

print\_matrix(matrix, rows, cols);

++cols;

for (int j = cols - 1; j > k; --j){

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

matrix[i][j] = matrix[i][j - 1];

}

}

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

matrix[i][k] = 0;

}

print\_matrix(matrix, rows, cols);

}

8 лаба

switch (menu) #include <iostream>

#include <fstream>

#include <Windows.h>

using namespace std;

const int M = 20, N = 20;

void shakerSort(float arr[], int n) {

bool swapped = true;

int start = 0;

int end = n - 1;

while (swapped) {

swapped = false;

for (int i = start; i < end; i++) {

if (arr[i] < arr[i + 1]) {

std::swap(arr[i], arr[i + 1]);

swapped = true;

}

}

if (!swapped) {

break;

}

swapped = false;

for (int i = end - 1; i >= start; i--) {

if (arr[i] < arr[i + 1]) {

std::swap(arr[i], arr[i + 1]);

swapped = true;

}

}

start++;

end--;

}

}

void get\_mas1(int arr[], int& n, string filename) {

ifstream file(filename);

file >> n;

if (file.is\_open()) {

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

file >> arr[i];

}

file.close();

}

else {

cout << "Unable to open file" << endl;

}

}

void show\_mas1(int arr[], int n, string filename) {

ofstream out(filename);

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

out << arr[i] << " ";

}

cout << endl;

}

void show\_mas3(float arr[], int n) {

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

cout << arr[i] << " ";

}

cout << endl;

}

void get\_mas2(int matrix[M][N], int& n, int& m, string filename) {

ifstream file(filename);

file >> m;

file >> n;

if (file.is\_open()) {

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

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

{

file >> matrix[i][j];

}

}

file.close();

}

else {

cout << "Unable to open file" << endl;

}

}

void get\_mas3(float arr[], int& n, string filename) {

ifstream file(filename);

file >> n;

if (file.is\_open()) {

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

file >> arr[i];

}

file.close();

}

else {

cout << "Unable to open file" << endl;

}

}

void task1() {

int n =0, t=0,fp=0;

int\* a = new int[n];

get\_mas1(a, n, "array\_in\_n.txt"); // передаем поток ifstream вместо имени файла

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

{

if (a[i] % 2 == 0 && t == 0)

{

fp = a[i];

t = 1;

}

else

{

if (a[i] % 2 == 0)

{

a[i] += fp;

}

}

}

show\_mas1(a, n, "array\_out\_n.txt"); // передаем поток ofstream вместо имени файла

delete[] a;

}

void task2() {

int matrix[M][N]{};

int Sum = 0, cols = 0, rows = 0,min\_max= -2147483648;//-2 147 483 648 минимальное значение типа int

get\_mas2(matrix, cols, rows, "matr\_in\_n.txt"); // передаем поток ifstream вместо имени файла

ofstream out("matr\_in\_n.txt", ios::app);

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

{

int min = matrix[rows][0];

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

{

if (j == 0 || matrix[i][j] < min)

{

min = matrix[i][j];

}

if (i == 0 || min > min\_max)

{

min\_max = min;

}

}

}

out << "Min max:" << min\_max << endl;

out.close();

}

void task3() {

int n = 0;

float\* arr = new float[n];

get\_mas3(arr, n, "shaker.txt");

cout << "Масив до сортування:" << endl;

show\_mas3(arr, n);

shakerSort(arr, n);

cout << "Масив після сортування:" << endl;

show\_mas3(arr, n);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int menu;

do

{

cout << "Task number(0 for exit):";

cin >> menu;

{

case 1:

task1();

break; // Завдання 1

case 2:

task2();

break; // Завдання 2

case 3:

task3();

break; // Завдання 3

}

} while (menu);

system("pause");

return 0;

}

9 лаба

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

using namespace std;

void myAppend(char str[], const char toAdd[]) {

int i = 0, j = 0;

while (str[i] != '\0') {

i++;

}

while (toAdd[j] != '\0') {

str[i] = toAdd[j];

i++;

j++;

}

str[i] = '\0';

}

void task1()

{

char str[50];

cout << "Enter a string: ";

cin.getline(str, 50);

string stdStr(str);

string toAdd = " is awesome!";

stdStr.append(toAdd);

myAppend(str, " C++");

myAppend(str, " is awesome!");

cout << "Using std::string: " << stdStr <<endl;

cout << "Using myAppend: " << str <<endl;

}

// Функція, що обробляє рядок S відповідно до завдання

string remove\_last\_substring(string S, string S0) {

size\_t pos = S.rfind(S0); // Знаходимо позицію останньої збігаючоїся підстроки

if (pos != string::npos) { // Якщо підстрока знайдена

S.erase(pos, S0.length()); // Видаляємо знайдену підстроку

}

return S;

}

// Функція, яка перевіряє, чи задовольняє рядок умовам завдання

bool is\_valid\_string(string S, string S0) {

return S.find(S0) != string::npos; // Перевіряємо, чи є збігаючі підстроки

}

void task2() {

string input\_filename = "input.txt"; // Ім'я вхідного файлу

string output\_filename = "output.txt"; // Ім'я вихідного файлу

string S0; // Рядок S0

cout << "Enter S0: ";

cin >> S0;

ifstream input\_file(input\_filename); // Відкриваємо вхідний файл

ofstream output\_file(output\_filename); // Відкриваємо вихідний файл

if (input\_file.is\_open() && output\_file.is\_open()) { // Перевіряємо, чи вдалося відкрити файли

string S;

while (getline(input\_file, S)) { // Зчитуємо рядок за рядком з вхідного файлу

if (is\_valid\_string(S, S0)) { // Перевіряємо, чи задовольняє рядок умовам

S = remove\_last\_substring(S, S0); // Обробляємо рядок

}

output\_file << S << endl; // Записуємо рядок у вихідний файл

}

input\_file.close(); // Закриваємо вхідний файл

output\_file.close(); // Закриваємо вихідний файл

}

else {

cout << "Error opening files" << endl;

}

}

int main()

{

task1();

task2();

return 0;

}

Лаба 10

#include <iostream>

using namespace std;

struct TPoint {

double x;

double y;

};

struct TTriangle {

TPoint A;

TPoint B;

TPoint C;

};

// Обчислення відстані між двома точками

double distance(TPoint A, TPoint B) {

return sqrt((B.x - A.x) \* (B.x - A.x) + (B.y - A.y) \* (B.y - A.y));

}

// Обчислення висоти трикутника, опущеної на відрізок AB

double height(TPoint A, TPoint B, TPoint C) {

return 2 \* abs((B.x - A.x) \* (C.y - A.y) - (C.x - A.x) \* (B.y - A.y)) / distance(A, B);

}

// Обчислення висот для всіх сторін трикутника

void Altitudes(TTriangle T, double& h1, double& h2, double& h3) {

h1 = height(T.A, T.B, T.C);

h2 = height(T.B, T.C, T.A);

h3 = height(T.C, T.A, T.B);

}

void task1() {

// Тестування Perim

TPoint A = { 1, 1 };

TPoint B = { 4, 5 };

TPoint C = { 2, 8 };

TTriangle ABC = { A, B, C };

TPoint D = { 7, 3 };

TTriangle ABD = { A, B, D };

TTriangle ACD = { A, C, D };

// Тестування Altitudes

double h1, h2, h3;

Altitudes(ABC, h1, h2, h3);

cout << "Altitudes of ABC: " << h1 << " " << h2 << " " << h3 << endl;

Altitudes(ABD, h1, h2, h3);

cout << "Altitudes of ABD: " << h1 << " " << h2 << " " << h3 << endl;

Altitudes(ACD, h1, h2, h3);

cout << "Altitudes of ACD: " << h1 << " " << h2 << " " << h3 << endl;

}

struct begin24 {

int A, B, C;

bool bool24() {

if (A!=B&&B!=C&&A!=C)

{

return true;

}

return false;

}

void swap() {

A = A + C;

C = A - C;

A = A - C;

B = B + A;

A = B - A;

B = B - A;

}

};

void task2() {

begin24 numbers{};

cout << "Enter A:";

cin >> numbers.A;

cout << "Enter B:";

cin >> numbers.B;

cout << "Enter C:";

cin >> numbers.C;

cout << "A:" << numbers.A << " B:" << numbers.B << " C:" << numbers.C << endl;

if (numbers.bool24())

{

numbers.swap();

cout <<"After swap"<<endl << "A:" << numbers.A << " B:" << numbers.B << " C:" << numbers.C << endl;

}

}

struct boolean24

{

int A, B, C;

bool bool24() {

if (A==0)

{ return false;

}

return true;

}

bool diskriminant()

{

double d =B \* B - 4 \* A\* C;

return d >= 0;

}

};

void task3() {

boolean24 obj{};

cout << "Enter A:";

cin >> obj.A;

cout << "Enter B:";

cin >> obj.B;

cout << "Enter C:";

cin >> obj.C;

if (obj.bool24())

{

if (obj.diskriminant())

{

cout << "Yes" << endl;

}

else

{

cout << "No" << endl;

}

}

else

{

cerr << "Error" << endl;

}

}

int main() {

task1();

task2();

task3();

return 0;

}

Лаба 11

namespace Project {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(-1, 95);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(282, 23);

this->button1->TabIndex = 0;

this->button1->Text = L"Порахувати";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 42);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(113, 16);

this->label1->TabIndex = 1;

this->label1->Text = L"Кількість секунд";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(-4, 164);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(156, 16);

this->label2->TabIndex = 2;

this->label2->Text = L" кількість повних годин";

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(178, 39);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(100, 22);

this->textBox1->TabIndex = 3;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(178, 161);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(100, 22);

this->textBox2->TabIndex = 4;

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(282, 253);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->button1);

this->Name = L"MyForm";

this->Text = L"MyForm";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int N = Convert::ToInt32(textBox1->Text);

if (N>=3600)

{

int M = N / 3600;

this->textBox2->Text = M.ToString();

}

else {

MessageBox::Show("Некоректні данні спробуйте знову.", "Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}