Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Иркутский национальный исследовательский технический

университет»

Институт информационных технологий и анализа данных

**О Т Ч Ё Т**

|  |  |  |
| --- | --- | --- |
| о прохождении | | учебной практики |
|  | | (вид практики: учебная/производственная) |
| технологической (проектно-технологической) практики | | |
| (тип практики: технологическая/научно-исследовательская работа/преддипломная и др.) | | |
|  | | |
| в | ИРНИТУ | |
|  | (наименование профильной организации) | |

Обучающегося Москвитина Михаил Алексеевича, ИСИб-24-1,

(ФИО, группа, подпись)

Руководитель практики от института ИТиАД

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Ссылка на резюме обучающего на сайте https://www.superjob.ru/

Руководитель образовательной программы

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)



Оценка по практике

(ФИО, подпись, дата)

Содержание отчета на \_\_\_ стр. Приложение к отчету на \_\_\_ стр.

Иркутск 2025

Ссылка на резюме обучающего на сайте https://www.hh.ru/

Оглавление

[Индивидуальное задание на прохождение учебной практики: технологической (проектно-технологической) практики 3](#_Toc209701481)

[Задание на практику №1 6](#_Toc209701482)

[Задание на практику № 2 8](#_Toc209701483)

[Задание на практику № 3 10](#_Toc209701484)

[Задание на практику № 4 12](#_Toc209701485)

[Задание на практику № 5 14](#_Toc209701486)

[Задание на практику № 6 15](#_Toc209701487)

[Задание на практику № 8 17](#_Toc209701488)

[Задание на практику № 9 20](#_Toc209701489)

[Задание на практику № 10 22](#_Toc209701490)

[Задание на практику № 11 24](#_Toc209701491)

[Заключение 26](#_Toc209701492)

# Индивидуальное задание на прохождение учебной практики: технологической (проектно-технологической) практики

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для | Москвитина Михаила Алексеевича | | | | |
|  | (ФИО обучающегося полностью) | | | | |
| обучающегося | | 1 | курса | группы | ИСИб-24-1 |

по направлению подготовки Информационные системы и технологии

профиль Интеллектуальные системы обработки информации и управления

Место прохождения практики: ИРНИТУ

Сроки прохождения практики с «16» июня 2025 г. по «29» июня 2025 г.

Цели и задачи прохождения практики: повысить свой уровень как разработчика и познакомиться с работой IT-индустрии. Прокачать навыки программирования через решение логических и практических задач. Поработать с электронными устройствами и реализовать собственные программные проекты. Увидеть изнутри, как устроены IT-компании, благодаря экскурсиям.

Содержание практики, вопросы, подлежащие изучению: Решение задач, направленных на укрепление своего опты в IT. Углубленное изучение виртуальных устройств.

Планируемые результаты практики: официальный отчёт о проделанной работе.

Руководитель практики от

института ИТиАД

\_\_\_\_\_\_\_\_\_\_\_/ Кононенко Р.В. /

(подпись

**Согласовано:**

Руководитель ООП

\_\_\_\_\_\_\_\_\_\_\_/Кононенко Р.В./

(подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2025 г.

С настоящим индивидуальным заданием и с программой практики ознакомлен, задание принято к исполнению

\_\_\_\_\_\_\_\_\_\_«16» июня 2025 г.

(подпись)

**ДНЕВНИК**

прохождения практики

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| обучающегося | | | | | Москвитина Михаила Алексеевича , ИСИб-24-1 |
|  | | | | | (фамилия, имя, отчество, группа) |
| курс | | 1 | | | |
| направление | | | | Информатика и вычислительная техника | |
| профиль | | | Интеллектуальные системы обработки | | |
| информации и управления | | | | | |
| в | ИРНИТУ | | | | |
|  | (наименование профильной организации) | | | | |
|  | Иркутск 2025 | | | | |

Руководителем практики от структурного подразделения назначен:

Кононенко Роман Владимирович, доцент

(ФИО, должность)

**Рабочий график (план) прохождения практической подготовки**

(заполняется обучающимся)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Период  практики | Содержание выполненных работ | Подпись руководителя практики от структурного подразделения |
| 1 | 16.06.2025 | Выполнена задача №1, задача №2 |  |
| 2 | 16.06.2025 | Выполнена задача №3 |  |
| 3 | 18.06.2025 | Создано резюме на сайте hh.ru, superjob.ru |  |
| 4 | 18.06.2025 | Выполнена задача №4, задача №5, задача №6 |  |
| 5 | 19.06.2025 | Выполнена задача №7, задача №8 |  |
| 6 | 24.06.2025 | Экскурсия по филиалу системного оператора единой энергетической системы РДУ |  |
| 7 | 24.06.2025 | Выполнена задача №9 |  |
| 8 | 25.06.2025 | Выполнена задача №10 |  |
| 9 | 26.06.2025 | Экскурсия по офису IT-компании “ISPsystem” |  |
| 10 | 26.06.2025 | Выполнена задача №11 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| Дата фактического прибытия |  |
| обучающегося в структурное подразделение | 16.06.2025 |
| Дата фактического убытия |  |
| обучающегося из структурного подразделения | 28.06.2025 |

|  |  |
| --- | --- |
| Руководитель образовательной программы | Кононенко Р.В. |
|  | (ФИО, подпись) |
| Директор института | Говорков А.С. |
|  | (ФИО, подпись) |

## Задание на практику №1

Незнайка в своей экспедиции на Луну оказался на вершине лунной горы. Спуск вниз опасен, поэтому он взял с собой карту склона горы, где числами обозначено, сколько минут требуется на этот участок маршрута. Спуск происходит сверху вниз на один из соседних участков. Пример наиболее короткого маршрута выделен красным цветом, сумма чисел = 10.



Рисунок 1

Напишите программу, рассчитывающую минимальное время спуска (сумму чисел в пути с вершины до основания).

**Код решения:**

**#include <iostream>**

**#include <vector>**

**#include <cstdlib>**

**#include <ctime>**

**using namespace std;**

**int main() {**

**srand(time(0));**

**cout << "Enter the height of the pyramid" << " ";**

**int N;**

**cin >> N;**

**vector<vector<int>> p(N), dp(N);**

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

**p[i].resize(i + 1); dp[i].resize(i + 1);**

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

**cout << (p[i][j] = dp[i][j] = rand() % 101) << " ";**

**}**

**cout << endl;**

**}**

**for (int i = N - 2; i >= 0; i--)**

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

**dp[i][j] += min(dp[i + 1][j], dp[i + 1][j + 1]);**

**cout << "\n" << dp[0][0] << "\n" << p[0][0] << " ";**

**for (int i = 1, j = 0; i < N; i++) {**

**j = (dp[i][j] < dp[i][j + 1]) ? j : j + 1;**

**cout << p[i][j] << " ";**

**}**

**}**

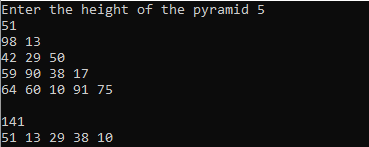
**Пример 1:  
**

Рисунок 2

**Пример 2:**

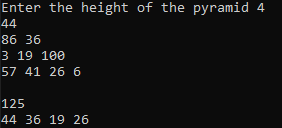


Рисунок 3

## Задание на практику № 2

После метеоритной атаки компьютерная сеть для управления лунными заводами разбилась на части, нужно объединить её в единое целое. Каждый фрагмент сети представлен в виде ненаправленного графа.

Вам известно общее число вершин графа (узлы сети, не более 1000) и набор рёбер (сохранившиеся линии связи, не более 1000).

Определите, какое минимальное число линий связи нужно дополнительно построить, чтобы сеть стала единой.



Рисунок 4



Рисунок 5

**Код решения (Python):**

def dfs(g, graph, visited):

visited[g] = True

for v in graph[g]:

if not visited[v]:

dfs(v, graph, visited)

def res():

N, M = map(int, input().split())

graph = {i: [] for i in range(1, N + 1)}

for \_ in range(M):

u, v = map(int, input().split())

graph[u].append(v)

graph[v].append(u)

visited = [False] \* (N + 1)

comp = 0

for i in range(1, N + 1):

if not visited[i]:

dfs(i, graph, visited)

comp += 1

print(comp - 1)

if \_\_name\_\_ == "\_\_main\_\_":

res()

**Пример 1:**

****

Рисунок 6

**Пример 2:**

****

Рисунок 7

## Задание на практику № 3

В Иркутске раз в году наступает зима. Не смотря на то что событие это довольно регулярное, оно всегда внезапно. Снег буквально заваливает все улицы, не давая проехать на чём-то меньше трактора. В этом году терпение лопнуло и специальным указом был создан кризисный центр по борьбе с сугробами. Центру были переданы спутники, лазеры, метеорологические зонды и несколько десятков лопат.

Вам поручено возглавить отдел разведки снежной ситуации и быть способным чрезвычайно быстро отвечать на запросы центра. Сам город состоит из нескольких, расположенных подряд, улиц, каждая из которых абсолютна похожа на любую другую.

* Информация о снеге передается вам в виде тройки чисел – 1 в качестве идентификатора события, уникального индекса улицы и количество миллиметров выпавшего снега.
* Запросы в свою очередь так же имеют вид тройки чисел – 2 в качестве идентификатора события, индекс улицы с которой нужно суммировать количество выпавшего снега и индекс улицы по которую нужно суммировать, крайние улицы должны быть включены.

**Код решения (c++):**

#include <iostream>

#include <vector>

using namespace std;

int main() {

    int n, k;

    cin >> n >> k;

    vector<long long> arr(n + 1, 0);

    while (k--) {

        int t, i, x;

        cin >> t;

        if (t == 1) {

            cin >> i >> x;

            arr[i] += x;

        } else {

            int u, r;

            cin >> u >> r;

            long long sum = 0;

            for (int j = u; j <= r; j++) sum += arr[j];

            cout << sum << endl;

        }

    }

}

Пример 1

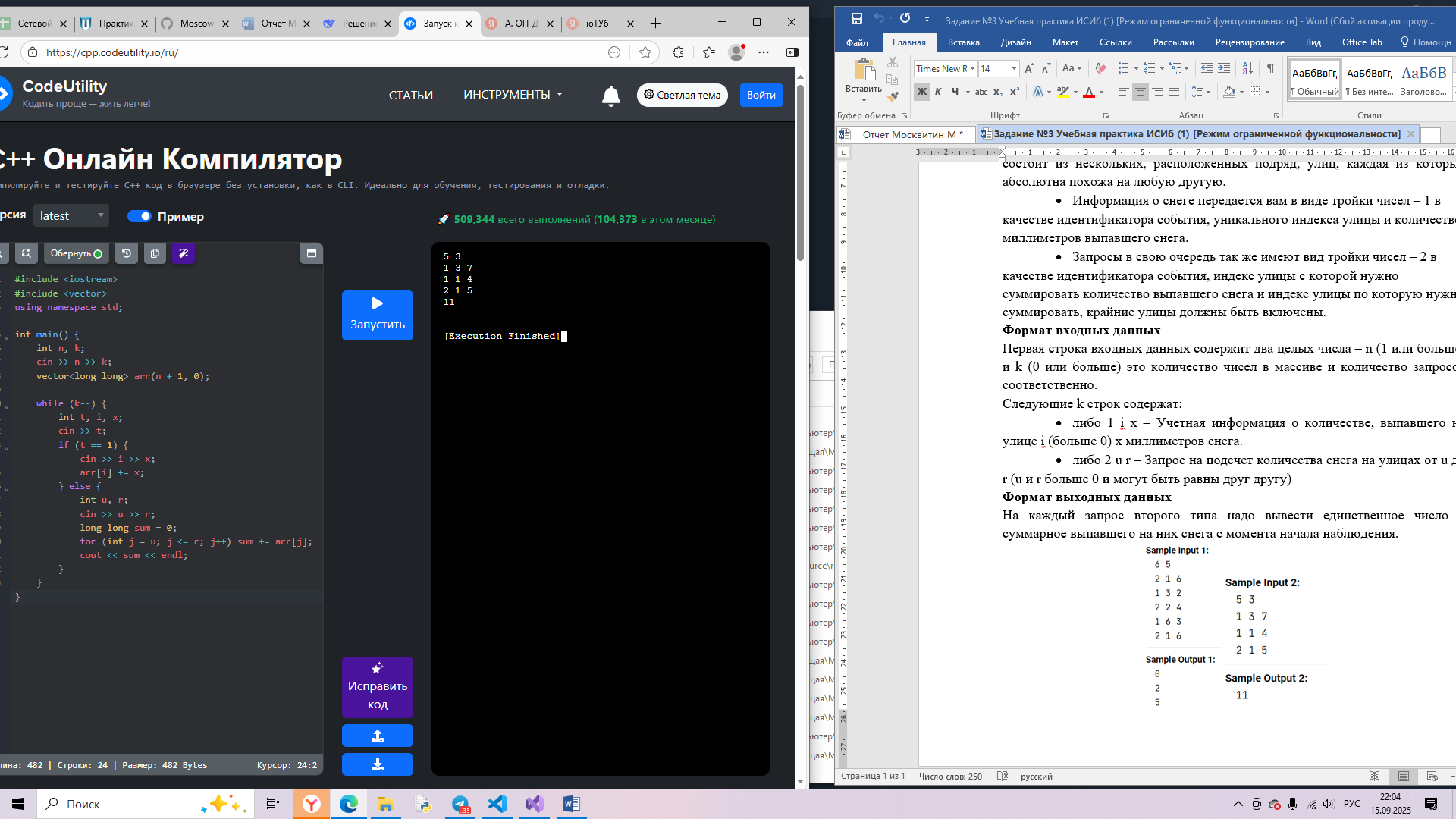


Рисунок 8

## Задание на практику № 4

Перестановка P длины n − это упорядоченный набор, содержащий числа от 1 до n, каждое из которых входит в него ровно один раз. Например, перестановкой длины 13 является набор (5 11 13 12 6 1 8 4 10 9 7 2 3). Само название говорит о том, для чего предназначен этот объект. Например, можно при помощи перестановки букв зашифровать слово. Для примера возьмем приведенную выше перестановку и слово transposition, которое состоит тоже из 13 букв. Далее, следуя перестановке, на первую позицию поставим пятую букву слова, на вторую − одиннадцатую букву и так далее. В итоге получим sinoptsntiora. К этому слову снова применим эту же перестановку и получим poartsnoitsin. Повторив эти стадии шифрования k раз, получим зашифрованное сообщение.



Рисунок 9

Вам дано зашифрованное таким образом слово, шифрующая перестановка P и число k. Необходимо восстановить слово.

**Код решения**(C++):

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

    int n, k;

    cin >> n >> k;

    vector<int> p(n);

    for (int i = 0; i < n; i++) {

        cin >> p[i];

        p[i]--;

    }

    string s;

    cin >> s;

    vector<int> cycle(n);

    for (int i = 0; i < n; i++) {

        int pos = i;

        for (int j = 0; j < k % n; j++) {

            pos = p[pos];

        }

        cycle[pos] = i;

    }

    string res(n, ' ');

    for (int i = 0; i < n; i++) {

        res[cycle[i]] = s[i];

    }

    cout << res << endl;

}

Пример 1:

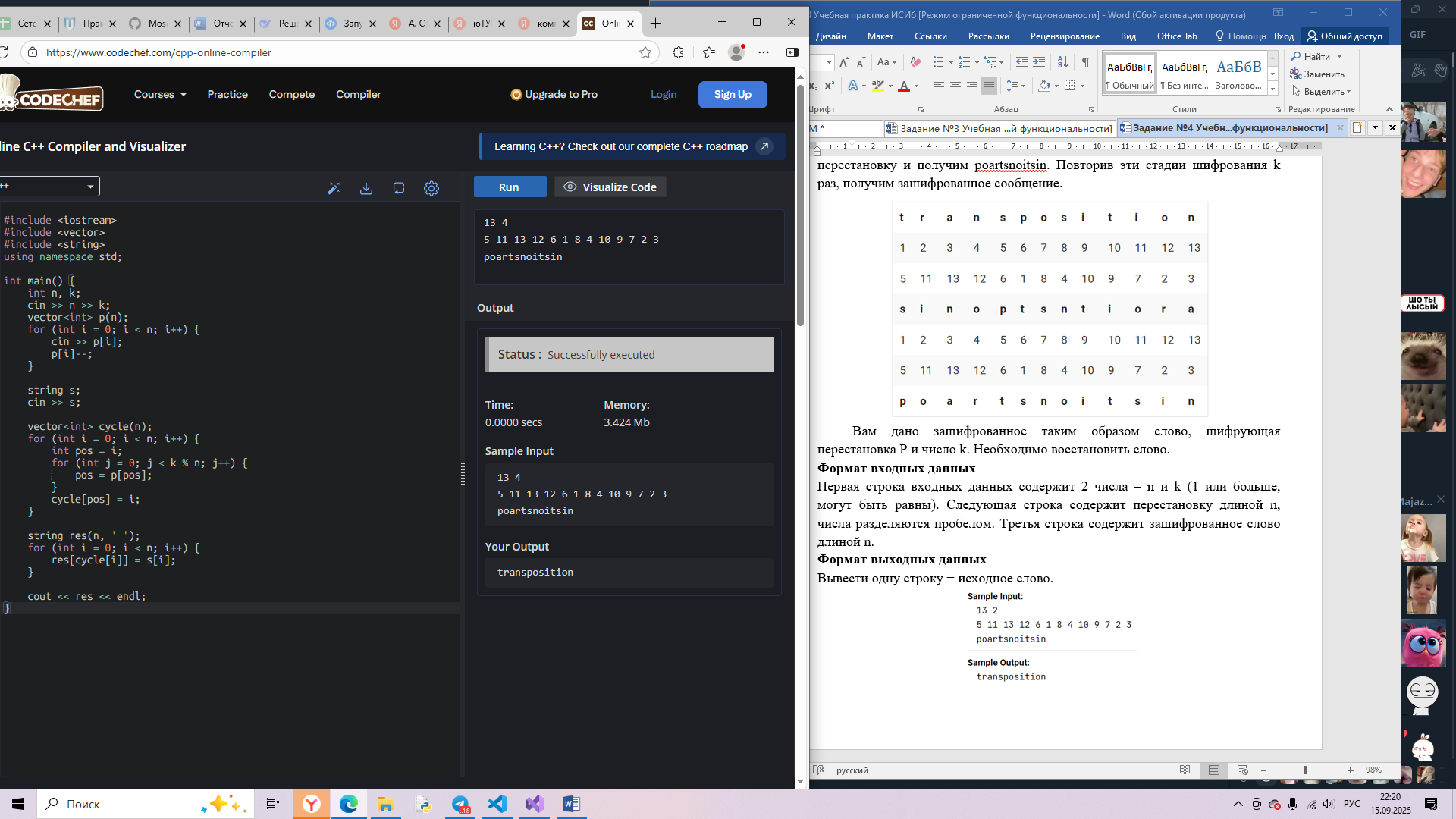


Рисунок 10

## Задание на практику № 5

Дана матрица, состоящая из 1 и 0. Значениями 1 в матрице нарисована некоторая фигура. Необходимо определить координаты верхнего левого и нижнего правого углов параллельного осям ограничивающего прямоугольника, т.е. такого прямоугольника, минимального размера, в который фигура помещается полностью и при этом ни одна точка исходной фигуры не попадает на стороны прямоугольника.

**Код решения** (C++):

#include <iostream>

using namespace std;

int main() {

    int h, w;

    cin >> h >> w;

    int min\_r = h, max\_r = -1;

    int min\_c = w, max\_c = -1;

    for (int i = 0; i < h; i++) {

        string s;

        cin >> s;

        for (int j = 0; j < w; j++) {

            if (s[j] == '1') {

                if (i < min\_r) min\_r = i;

                if (i > max\_r) max\_r = i;

                if (j < min\_c) min\_c = j;

                if (j > max\_c) max\_c = j;

            }

        }

    }

    cout << min\_r << " " << min\_c << " " << max\_r << " " << max\_c;

}

Пример 1:

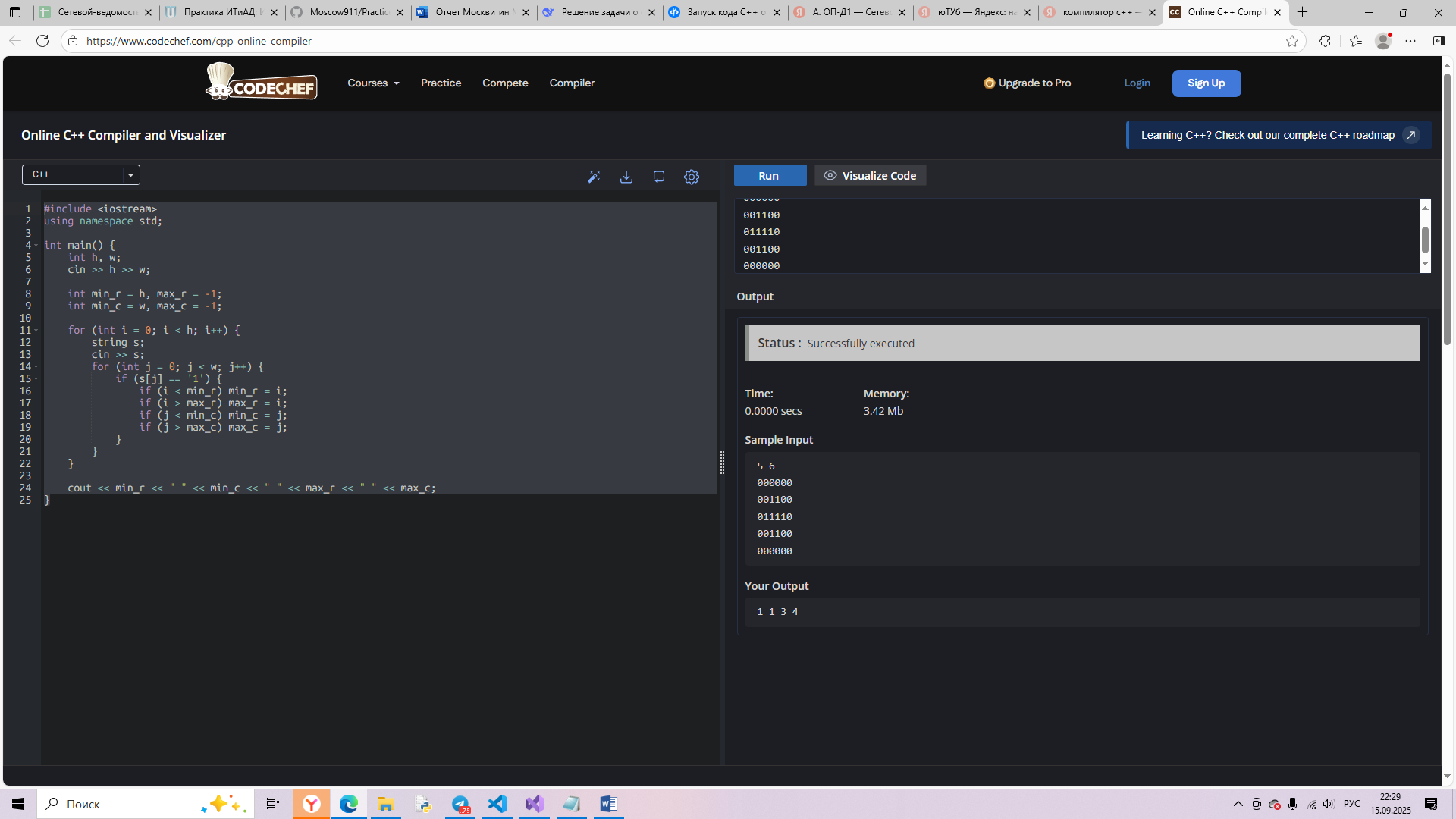


Рисунок 11

## Задание на практику № 6

В школьном кружке робототехники есть два вида микроконтроллеров (условно тип A и тип B) и два вида модулей управления мотором (условно тип 1 и тип 2). Выяснилось, что контроллер типа B и модуль управления типа 2 несовместимы. Использование микроконтроллеров и модулей управления в других комбинациях возможно. Имеется a микроконтроллеров типа A, b микроконтроллеров типа B, x модулей управления типа 1 и y модулей типа 2. Определите, какое максимальное число работающих пар из микроконтроллера и модуля управления мотором можно составить. Ваша программа должна ответить на n запросов.

**Код решения**(C++):

#include <iostream>

using namespace std;

int main() {

    int n;

    cin >> n;

    for (int i = 0; i < n; i++) {

        int a, b, x, y;

        cin >> a >> b >> x >> y;

        int pairs = min(a, x) + min(a, y) + min(b, x);

        cout << pairs << " ";

    }

}

Пример 1:

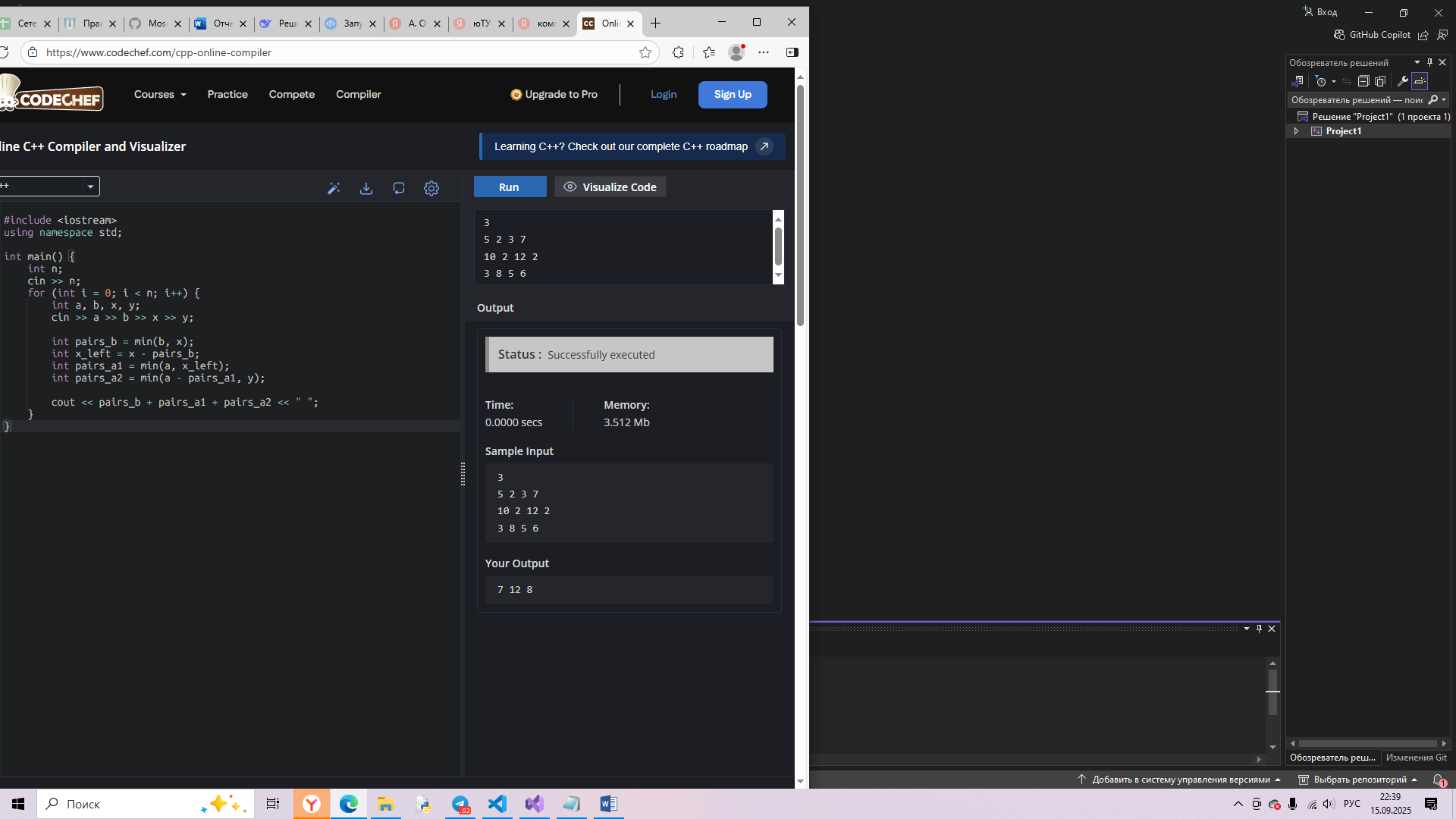


Рисунок 12

**Задание на практику № 7**

На компьютере работника автосервиса нашли файл с последовательностью автомобильных номеров, обслуживавшихся в этом автосервисе. Так как файл был поврежден, некоторые данные отображаются неверно. Нужно определить, какие из них остались невредимыми.

Автомобильным номером является строка из шести символов. Первый символ – заглавная латинская буква, далее следует 3 цифры, и после – две заглавные латинские буквы. Например, строка "P142EQ" является номером. Вам будет дана строка, состоящая из шести символов, необходимо ответить, является ли строка автомобильным номером.

Код решения(C++):

#include <iostream>

#include <string>

using namespace std;

int main() {

    string s;

    cin >> s;

    if (s.size() != 6) {

        cout << "No";

        return 0;

    }

    if (s[0] >= 'A' && s[0] <= 'Z' &&

        s[1] >= '0' && s[1] <= '9' &&

        s[2] >= '0' && s[2] <= '9' &&

        s[3] >= '0' && s[3] <= '9' &&

        s[4] >= 'A' && s[4] <= 'Z' &&

        s[5] >= 'A' && s[5] <= 'Z') {

        cout << "Yes";

    } else {

        cout << "No";

    }

}

Пример 1:

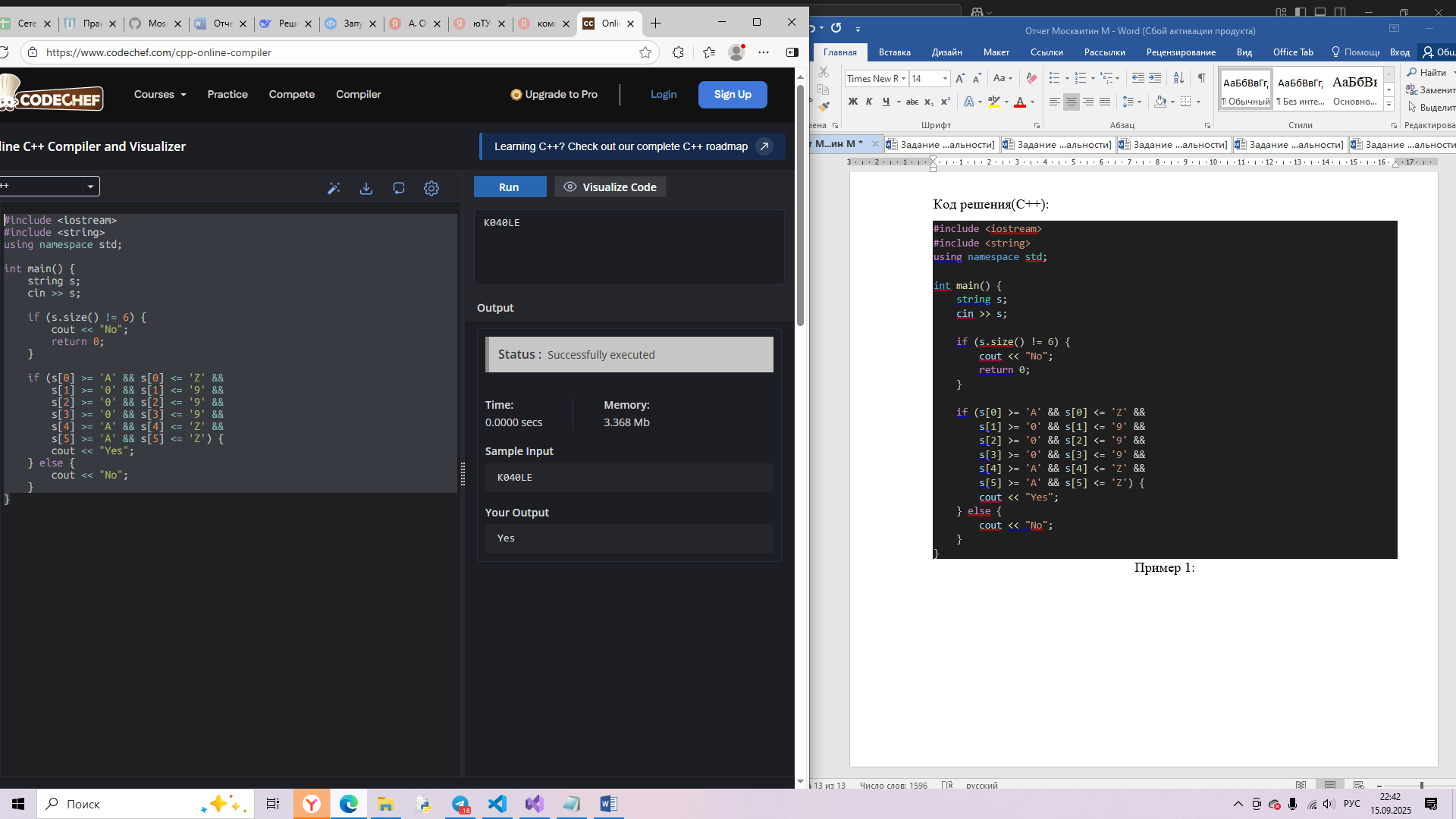


Рисунок 13

## Задание на практику № 8

Составить светодиодную матрицу размером не менее 8 на 8 светодиодов  
(пример на рисунке ниже размером 4 на 4)  
На матрицу вывести инфографику с различными динамично  
меняющимися изображениями.

Код решения:

#include <FastLED.h>

#define PIN 3

#define NUMPIXELS 64

#define WIDTH 8

#define HEIGHT 8

CRGB leds[NUMPIXELS];

void setup() {

    FastLED.addLeds<NEOPIXEL, PIN>(leds, NUMPIXELS);

    FastLED.setBrightness(50);

}

void loop() {

    snakeRowsAlternatingWithFullTail();

}

int getSnakeIndex(int x, int y) {

    if (y % 2 == 0) {

        return y \* WIDTH + x;

    } else {

        return y \* WIDTH + (WIDTH - 1 - x);

    }

}

void snakeRowsAlternatingWithFullTail() {

    const int snakeLength = 5;

    fill\_solid(leds, NUMPIXELS, CRGB::Black);

    for (int y = 0; y < HEIGHT; y++) {

        int start, end, step;

        if (y % 2 == 0) {

            start = WIDTH - 1;

            end = -snakeLength;

            step = -1;

        } else {

            start = 0;

            end = WIDTH + snakeLength - 1;

            step = 1;

        }

        for (int head = start; (step == 1 ? head < end : head > end); head += step) {

            fill\_solid(leds, NUMPIXELS, CRGB::Black);

            for (int k = 0; k < snakeLength; k++) {

                int x = (step == 1) ? head - k : head + k;

                if (x >= 0 && x < WIDTH) {

                    int idx = getSnakeIndex(x, y);

                    leds[idx] = CRGB::Red;

                }

            }

            FastLED.show();

            delay(200);

        }

    }

}

Пример 1:

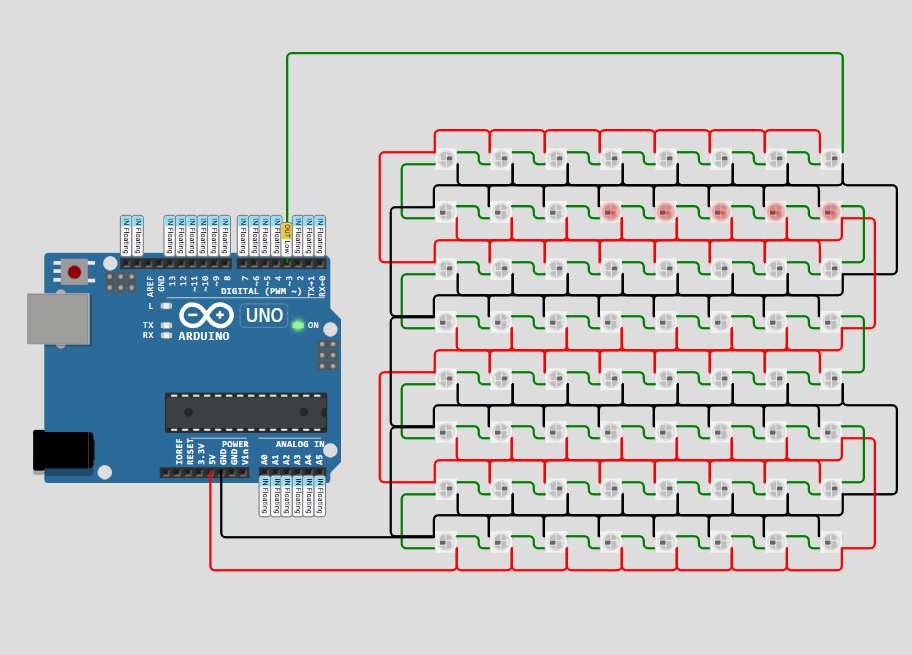


Рисунок 14

Пример 2:

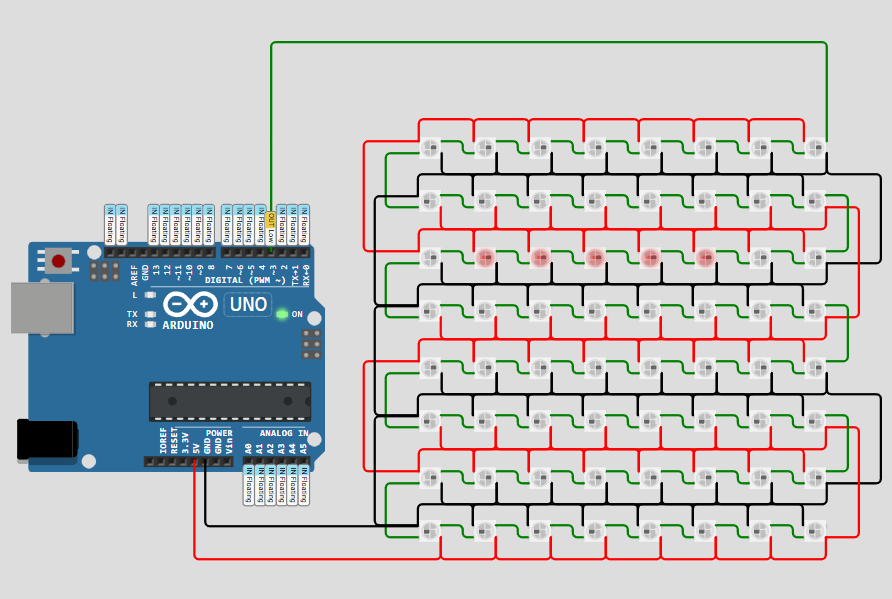


Рисунок 15

## Задание на практику № 9

Задачи:  
1) Собрать схему имитирующую работу автоматических дверей  
2) Подобрать номинал резисторов для светодиодов  
3) Написать программу для управления процессом работы автоматических  
дверей.  
Схема приведена на рисунке 1.  
Зеленый светодиод – двери отрываются.  
Красный светодиод – двери закрываются.  
Фоторезистор имитируют процесс приближения-удаления человека от дверей.

Код решения:

const int photoresistorPin = A0;

const int redLedPin = 9;

const int greenLedPin = 5;

const int threshold = 512;

const unsigned long openDuration = 5000;

void setup() {

  pinMode(redLedPin, OUTPUT);

  pinMode(greenLedPin, OUTPUT);

**Serial**.begin(9600);

  digitalWrite(redLedPin, HIGH);

  digitalWrite(greenLedPin, LOW);

}

void loop() {

  int sensorValue = analogRead(photoresistorPin);

**Serial**.print("Sensor Value: ");

**Serial**.println(sensorValue);

  if (sensorValue > threshold) {

    digitalWrite(greenLedPin, HIGH);

    digitalWrite(redLedPin, LOW);

**Serial**.println("Двери открыты");

    delay(openDuration);

    sensorValue = analogRead(photoresistorPin);

    if (sensorValue > threshold) {

**Serial**.println("Двери остаются открытыми");

    } else {

      digitalWrite(greenLedPin, LOW);

      digitalWrite(redLedPin, HIGH);

**Serial**.println("Двери закрыты");

    }

  }

}

Пример 1:

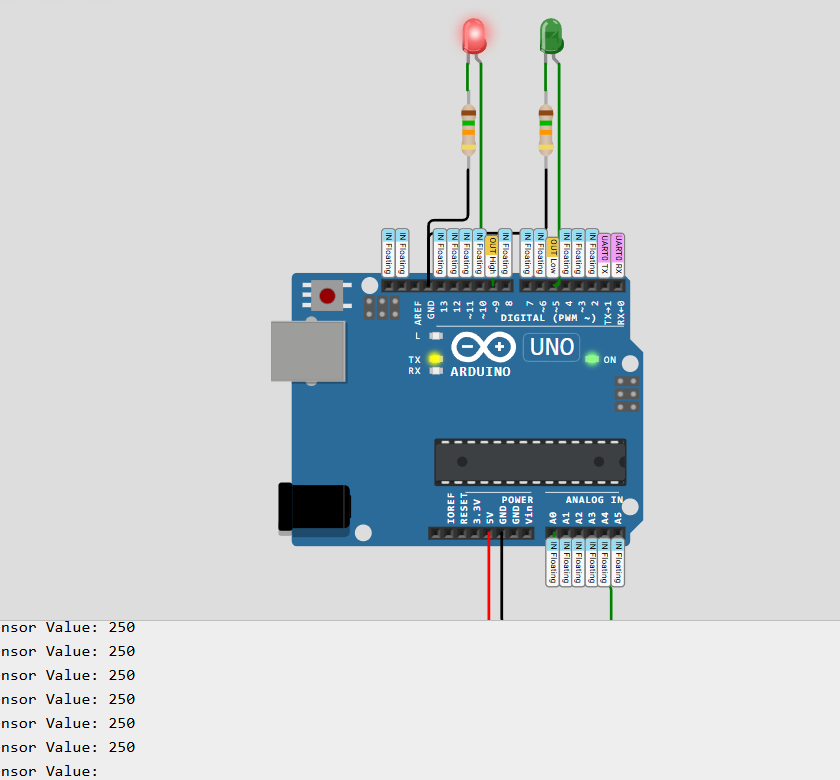


Рисунок 16

## Задание на практику № 10

Задачи:  
1) Собрать схему подключения сервопривода  
2) Написать программу для управления сервоприводом через  
последовательный порт

Код решения:

#include <Servo.h>

Servo myServo;

int currentAngle = 90;

int servoPin = 3;

void settings() {

**Serial**.begin(9600);

  myServo.attach(servoPin);

  myServo.write(currentAngle);

**Serial**.println("Введите угол от 0 до 180");

}

void check() {

  if (**Serial**.available() > 0) {

    String input = **Serial**.readStringUntil('\n');

    input.trim();

    if (isNumber(input)) {

      int targetAngle = input.toInt();

      if (targetAngle < 0 || targetAngle > 180) {

**Serial**.println("Ошибка");

      } else if (targetAngle != currentAngle) {

        move(targetAngle);

        currentAngle = targetAngle;

**Serial**.print("Установлен угол: ");

**Serial**.println(currentAngle);

      }

    } else {

**Serial**.println("Ошибка");

    }

**Serial**.println("Введите угол от 0 до 180:");

  }

}

void move(int target) {

  int step = (target > currentAngle) ? 1 : -1;

  for (int pos = currentAngle; pos != target; pos += step) {

    myServo.write(pos);

    delay(10);

  }

  myServo.write(target);

}

void setup() {

  settings();

}

void loop() {

  check();

}

bool isNumber(String str) {

  for (unsigned int i = 0; i < str.length(); i++) {

    if (!isDigit(str.charAt(i))) return false;

  }

  return str.length() > 0;

}

Пример 1:

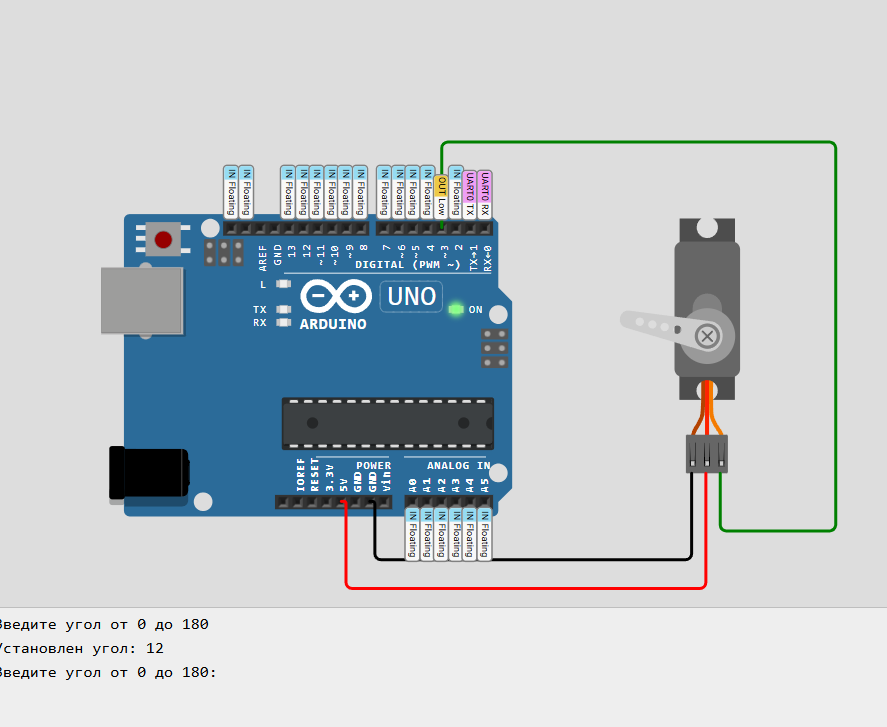


Рисунок 17

## Задание на практику № 11

Задание:

**Найти самый большой объект на изображении. Определить центр самого большого объекта. Обвести самый большой объект красной рамкой.**

**Код решения:**

**# Очень часто изображения обрабатываются как массив RGB-пикселей. Хотя эта идея кажется**

**# относительно интуитивной, она не оптимальна для решаемой нами задачи.**

**# В RGB цвет пикселя определяется насыщенностью красного, зеленого и синего.**

**# Таким образом, выбор диапазона оттенков одного цвета становится не самой простой задачей.**

**# Иначе обстоит дело с форматом HSV. Эта цветовая схема определяется тремя компонентами:**

**# Hue - цветовой тон;**

**# Saturation - насыщенность;**

**# Value - яркость.**

**# В схеме HSV базовый цвет можно выбрать с помощью компоненты Hue (например, красный, оранжевый и т.д.).**

**# Две другие компоненты позволяют регулировать насыщенность и яркость базового цвета,**

**# делая его более насыщенным или блеклым, светлее или темнее.**

**# Эти свойства HSV позволяют легко определять диапазоны, которые могут захватывать области нужного цвета и его оттенков.**

**import cv2**

**import numpy as np**

**# Указываем полный путь к файлу (изображению) или**

**# имя файла, если он сохранен в папке проекта**

**image = cv2.imread("11zadan.jpg")**

**# Создаем окно. Выводим изображение. Подписываем окно**

**cv2.imshow("original", image)**

**# Ждем нажатия клавиши для перехода к следующей команде**

**cv2.waitKey(0)**

**# Библиотека Open CV может делать различные операции с изображением**

**# Добавляем размытие для подавления шумов**

**blurred\_image = cv2.GaussianBlur(image, (11, 11), 0)**

**# Создаем окно. Выводим изображение. Подписываем окно**

**cv2.imshow("blurred", blurred\_image)**

**# Ждем нажатия клавиши для перехода к следующей команде**

**cv2.waitKey(0)**

**# Изменяем цветовую палитру (конвертируем RGB в HSV)**

**hsv\_image = cv2.cvtColor(blurred\_image, cv2.COLOR\_BGR2HSV)**

**# Создаем окно. Выводим изображение. Подписываем окно**

**cv2.imshow("hsv", hsv\_image)**

**# Ждем нажатия клавиши для перехода к следующей команде**

**cv2.waitKey(0)**

**# Устанавливаем границы цвета для зеленого**

**hsv\_min = np.array((36, 25, 25), np.uint8) # Нижняя граница зеленого цвета**

**hsv\_max = np.array((70, 255, 255), np.uint8) # Верхняя граница зеленого цвета**

**# Создаем маску с границами. Выделяем область**

**green\_mask = cv2.inRange(hsv\_image, hsv\_min, hsv\_max)**

**# Создаем окно. Выводим изображение. Подписываем окно**

**cv2.imshow("mask", green\_mask)**

**# Ждем нажатия клавиши для перехода к следующей команде**

**cv2.waitKey(0)**

**# Метод findContours находит контуры и возвращает их вместе с иерархией**

**# передаем копию объекта; алгоритм поиска; алгоритм аппроксимации контуров**

**contours, hierarchy = cv2.findContours(green\_mask.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)**

**# Сортируем все найденные контуры по убыванию площади**

**sorted\_contur = sorted(contours, key=cv2.contourArea, reverse=True)**

**# Find the largest object - берем первый (самый большой) контур**

**if len(sorted\_contur) > 0:**

**# Determine the coordinates of the largest object**

**x, y, w, h = cv2.boundingRect(sorted\_contur[0])**

**print(f"Largest object: x={x}, y={y}, width={w}, height={h}")**

**# Determine the center of the largest object**

**center\_x = x + w // 2**

**center\_y = y + h // 2**

**print(f"Center coordinates: ({center\_x}, {center\_y})")**

**# Outline the largest object with a red frame**

**cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 3)**

**# Draw the center point**

**cv2.circle(image, (center\_x, center\_y), 5, (0, 0, 255), -1)**

**# Add text with coordinates**

**cv2.putText(image, f"Center: ({center\_x}, {center\_y})",**

**(x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 2)**

**else:**

**print("No objects found")**

**# Создаем окно. Выводим изображение. Подписываем окно**

**cv2.imshow('Largest Object Detection', image)**

**# Ждем нажатия клавиши для перехода к следующей команде**

**cv2.waitKey(0)**

**cv2.destroyAllWindows()**

Пример 1:

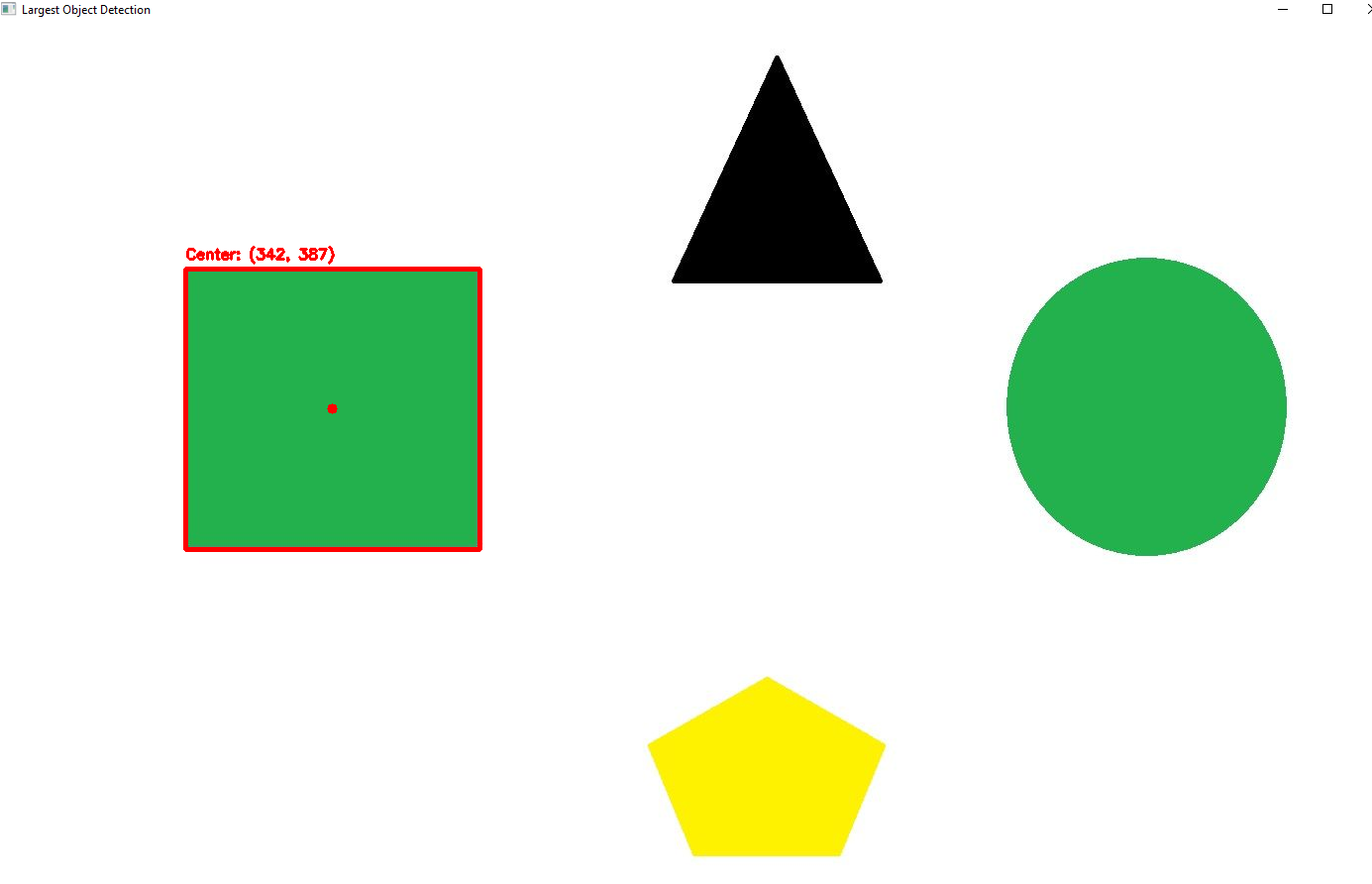


Рисунок 18

## Заключение

В ходе учебной практики я приобрел значительный опыт в области программирования. Решение разнообразных логических и алгоритмических задач способствовало более глубокому усвоению теоретического материала. Особый интерес представила работа с виртуальными электронными устройствами на платформе Arduino, где мы разрабатывали и запускали функциональные схемы.

Практика также включила две познавательные экскурсии. В филиале иркутского РДУ мы ознакомились с энергетической инфраструктурой региона и её взаимодействием с IT-технологиями. В компании ISPsystem сотрудники поделились опытом о реальной работе в сфере разработки ПО, дали ценные советы по карьерному развитию и трудоустройству.

Все поставленные цели практики были не только достигнуты, но и превышены, что делает её исключительно продуктивной.