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 Кузнецова Ильи Дмитриевича, ИСИб-24-1

(ФИО, группа, подпись)

Руководитель практики от института ИТиАД

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Ссылка на резюме на сайте superjob.ru

Руководитель образовательной программы

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)



Оценка по практике

(ФИО, подпись, дата)

Содержание отчета на 26 стр. Приложение к отчету на \_\_\_ стр.

Иркутск 2025

Ссылка на резюме на сайте hh.ru

**Индивидуальное задание на прохождение**

**учебной практики: технологической (проектно-технологической) практики**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для | Кузнецова Ильи Дмитриевича | | | | |
|  | (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: улучшить существующие навыки программирования, расширить способности в сфере разработки ПО, ближе познакомиться с деятельностью, задачами и особенностями IT организаций.

Содержание практики, вопросы, подлежащие изучению: логические задачи на программирование, задачи с виртуальными электронными устройствами, задача на реализацию программного обеспечения, поездки на экскурсии в компании, которые связанны с IT.

Планируемые результаты практики: отчётный документ по итогам практики по стандартам организации.

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 | | | | | Кузнецова Ильи Дмитриевича , 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 | 17.06.2025 | Выполнена задача №3 |  |
| 3 | 17.06.2025 | Создано резюме на сайте hh.ru, superjob.ru |  |
| 4 | 18.06.2025 | Выполнена задача №4, задача №5 |  |
| 5 | 19.06.2025 | Выполнена задача №6, задача №7 |  |
| 6 | 20.06.2025 | Выполнена задача №8 |  |
| 7 | 24.06.2025 | Экскурсия по филиалу системного оператора единой энергетической системы РДУ |  |
| 8 | 24.06.2025 | Выполнена задача №9 |  |
| 9 | 25.06.2025 | Выполнена задача №10 |  |
| 10 | 26.06.2025 | Экскурсия по офису IT-компании ISPsystem |  |
| 11 | 27.06.2025 | Выполнена задача №11 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| Дата фактического прибытия |  |
| обучающегося в структурное подразделение | 16.06.2026 |
| 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 №1**

Постановка задачи:

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

Ход решения:

import random

n = int(input())

mount = []

for r in range(n):

row = []

for c in range(r + 1):

row.append(random.randint(10, 100))

mount.append(row)

for row in mount:

print(row)

mount\_dp = []

for row in mount:

mount\_dp.append(row[:])

for r in range(n - 2, -1, -1):

for c in range(len(mount\_dp[r])):

mount\_dp[r][c] += min(mount\_dp[r+1][c], mount\_dp[r+1][c+1])

mint = mount\_dp[0][0]

print(mint)

p = []

r = 0

c = 0

p.append(mount[r][c])

while r < n - 1:

left = mount\_dp[r+1][c]

right = mount\_dp[r+1][c+1]

if mount\_dp[r+1][c] < mount\_dp[r+1][c+1]:

c += 0

else:

c += 1

r += 1

p.append(mount[r][c])

print(\*(p))

Результат:

5

[18]

[87, 94]

[52, 40, 48]

[15, 61, 69, 98]

[78, 39, 96, 74, 52]

211

18 87 52 15 39

Пример выполнения программы №1

3

[43]

[48, 29]

[22, 80, 60]

113

43 48 22

Пример выполнения программы №2

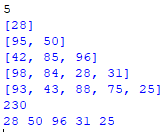


Рисунок 1 - Пример выполнения программы №3

**Задание №2**

Постановка задачи:

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.

Ход решения:

Код на 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()

Результат:

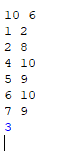


Рисунок 1 - Пример выполнения программы №1

**Задача №3**

Постановка задачи:

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.

Ход решения:

def f(snow, start, finish):

res = 0

for i in range(start, finish + 1):

res += snow[i]

return res

n, k = map(int, input().split())

snow = [0] \* (n + 1)

for \_ in range(k):

o = list(map(int, input().split()))

otype = o[0]

if otype == 1:

st, snowk = o[1], o[2]

snow[st] += snowk

elif otype == 2:

start, end = o[1], o[2]

res = f(snow, start, end)

print(res)

Результат:



Рисунок 1 - Пример выполнения программы №1

6 5

2 1 6

1 3 2

2 2 4

1 6 3

2 1 6

0

2

5

Пример выполнения программы №2

**Задание №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. Вам дано зашифрованное таким образом слово, шифрующая перестановка P и число k. Необходимо восстановить слово.

Ход решения:

def f(p):

n = len(p)

decode = [0] \* n

for i in range(n):

decode[p[i] - 1] = i

return decode

n, k = map(int, input().split())

p = list(map(int, input().split()))

a = input()

decode = f(p)

for \_ in range(k):

a = ''.join(a[decode[i]] for i in range(n))

print(a)

Результат:

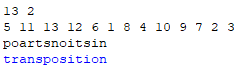
****

Рисунок 1 - Пример выполнения программы №1

****

Рисунок 2 - Пример выполнения программы №2

**Задание №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.

Ход решения:

h, w = map(int, input().split())

matrix = [list(map(int, input().split())) for \_ in range(h)]

min1\_r, max1\_r = h, -1

min1\_c, max1\_c = w, -1

for i in range(h):

for j in range(w):

if matrix[i][j] == 1:

min1\_r = min(min1\_r, i)

max1\_r = max(max1\_r, i)

min1\_c = min(min1\_c, j)

max1\_c = max(max1\_c, j)

topleft\_r = min1\_r - 1

topleft\_c = min1\_c - 1

lowerright\_r = max1\_r + 1

lowerright\_c = max1\_c + 1

print(topleft\_r, topleft\_c, lowerright\_r, lowerright\_c)

Результат:

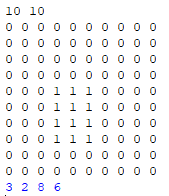
****

Рисунок 1 - Пример выполнения программы №1

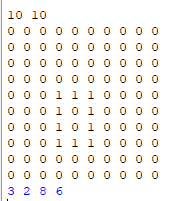
****

Рисунок 2 - Пример выполнения программы №2

**Задание №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.

Ход решения:

n = int(input())

res = []

for \_ in range(n):

a, b, x, y = map(int, input().split())

pairA2 = min(a, y)

have\_a = a - pairA2

have\_y = y - pairA2

pairA1 = min(have\_a, x)

have\_x = x - pairA1

pairB1 = min(b, have\_x)

pairs = pairA2 + pairA1 + pairB1

res.append(str(pairs))

print(' '.join(res))

Результат:



Рисунок 1 - Пример выполнения программы №1



Рисунок 2 - Пример выполнения программы №2

**Задание №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.

Ход решения:

letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

digits = "0123456789"

a = input()

if (len(a) == 6 and

a[0] in letters and

a[1] in digits and

a[2] in digits and

a[3] in digits and

a[4] in letters and

a[5] in letters):

print("Yes")

else:

print("No")

Результат:



Рисунок 1 - Пример выполнения программы №1



Рисунок 2 - Пример выполнения программы №2

**Задание №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.

Ход решения:

#include <FastLED.h>

#define PIN 3

#define WIDTH 8

#define HEIGHT 8

#define NUMPIXELS (WIDTH \* HEIGHT)

CRGB leds[NUMPIXELS];

const uint8\_t smileFace[8][8] = {

    {0,0,1,1,1,1,0,0},

    {0,1,0,0,0,0,1,0},

    {1,0,1,0,0,1,0,1},

    {1,0,0,0,0,0,0,1},

    {1,0,0,1,1,0,0,1},

    {1,0,1,0,0,1,0,1},

    {0,1,0,0,0,0,1,0},

    {0,0,1,1,1,1,0,0}

};

const uint8\_t sadFace[8][8] = {

    {0,0,1,1,1,1,0,0},

    {0,1,0,0,0,0,1,0},

    {1,0,1,0,0,1,0,1},

    {1,0,0,0,0,0,0,1},

    {1,0,1,0,0,1,0,1},

    {1,0,0,1,1,0,0,1},

    {0,1,0,0,0,0,1,0},

    {0,0,1,1,1,1,0,0}

};

void setup() {

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

    FastLED.setBrightness(50);

}

void showFace(const uint8\_t face[8][8], CRGB color = CRGB::Red) {

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

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

        for (int x = 0; x < WIDTH; x++) {

            if (face[y][x]) {

                int idx = y \* WIDTH + x;

                leds[idx] = color;

            }

        }

    }

    FastLED.show();

}

void loop() {

    showFace(smileFace, CRGB::Red);

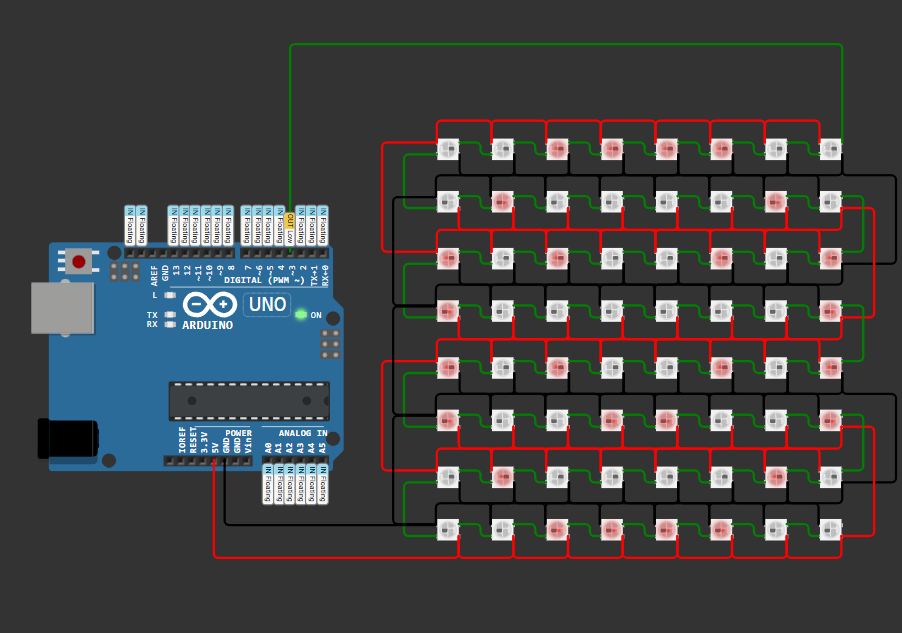
    delay(1000);

    showFace(sadFace, CRGB::Red);

    delay(1000);

}

Результат:



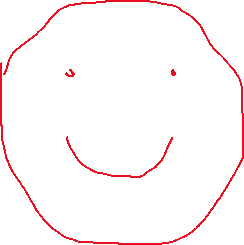
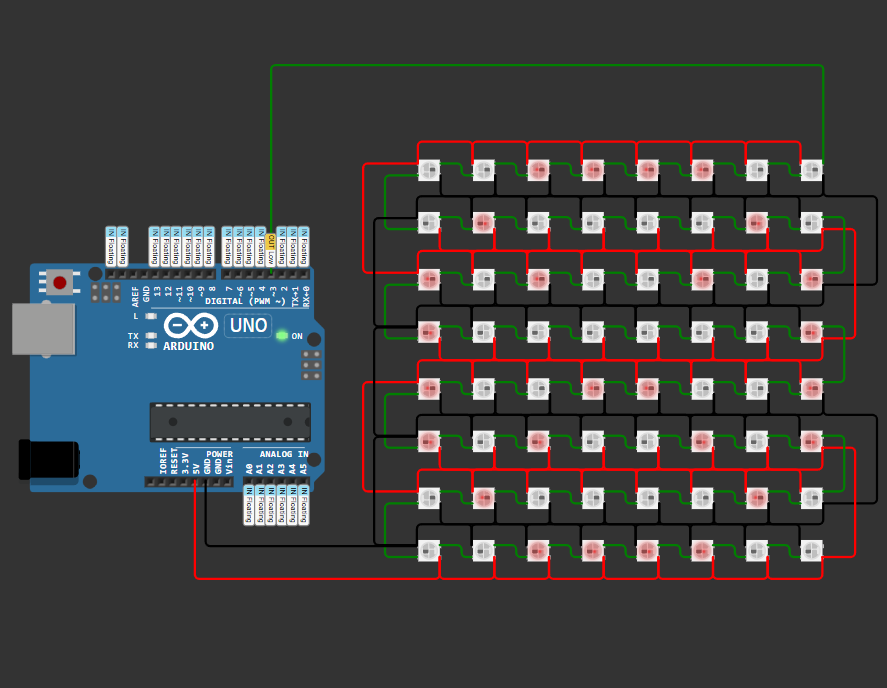


Рисунок 1 - Пример выполнения программы №1



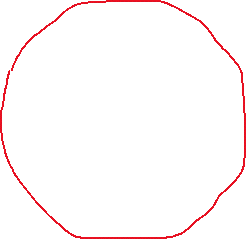


Рисунок 2 - Пример выполнения программы №2

**Задание №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.

Ход 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");

    }

  }

}

Результат:

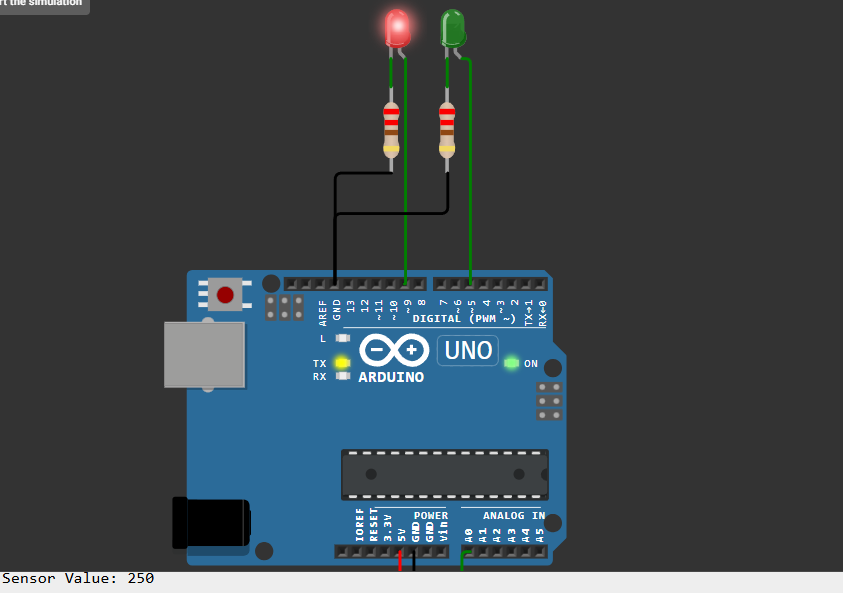
****

Рисунок 1 - Пример выполнения программы (для значения 250) №1

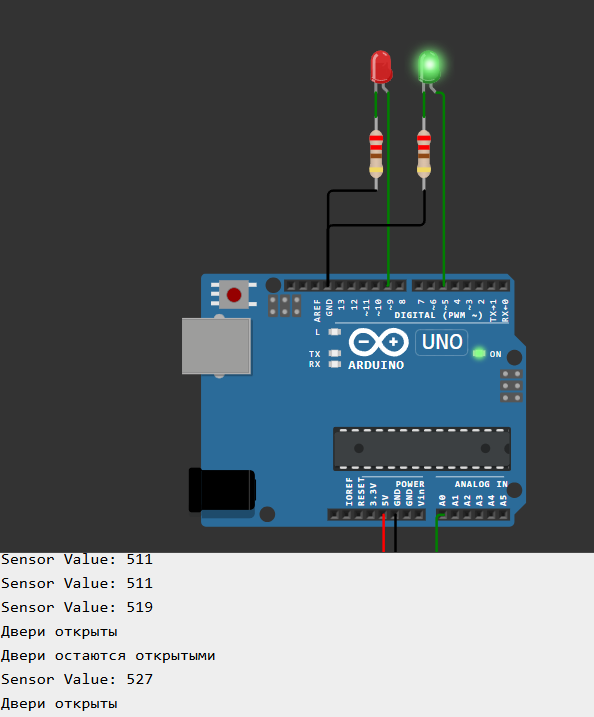


Рисунок 2 - Пример выполнения программы (для значения 527) №2

**Задание №10**

**Задачи:**

1. Собрать схему подключения сервопривода
2. Написать программу для управления сервоприводом через последовательный порт

Ход 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;

}

Результат:

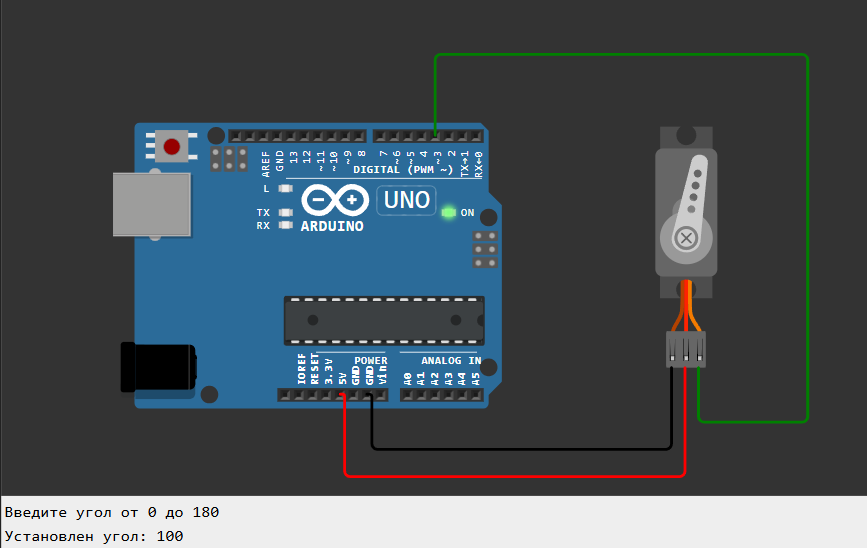


Рисунок 1 - Пример выполнения программы (для значения 100) №1

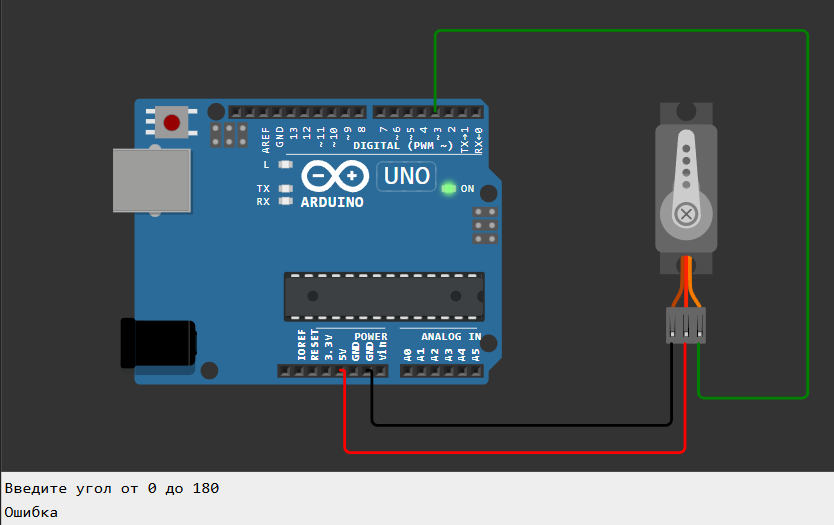


Рисунок 2 - Пример выполнения программы №2

**Задание №11**

**Задачи:**

Написать программу по поиску всех синих объектов на изображении. Найте центры синих объектов. Отметить центр красной точкой.

Ход решения:

import cv2  
import numpy as np  
  
image = cv2.imread("test2.jpg")  
cv2.imshow("original", image)  
cv2.waitKey(0)  
  
blurred\_image = cv2.GaussianBlur(image, (11, 11), 0)  
cv2.imshow("blurred", blurred\_image)  
cv2.waitKey(0)  
  
hsv\_image = cv2.cvtColor(blurred\_image, cv2.COLOR\_BGR2HSV)  
cv2.imshow("hsv", hsv\_image)  
cv2.waitKey(0)  
  
hsv\_min = np.array((90, 50, 50), np.uint8)  
hsv\_max = np.array((130, 255, 255), np.uint8)  
  
blue\_mask = cv2.inRange(hsv\_image, hsv\_min, hsv\_max)  
cv2.imshow("blue\_mask", blue\_mask)  
cv2.waitKey(0)  
  
contours, hierarchy = cv2.findContours(blue\_mask.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  
sorted\_contours = sorted(contours, key=cv2.contourArea, reverse=True)  
  
for i, contour in enumerate(sorted\_contours):  
 if cv2.contourArea(contour) > 100:  
 x, y, w, h = cv2.boundingRect(contour)  
 print(f"{x}, {y}, {w}, {h}")  
 center\_x = int(x + w/2)  
 center\_y = int(y + h/2)  
 cv2.circle(image, (center\_x, center\_y), 10, (0, 0, 255), -1)  
 cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2  
  
cv2.imshow('result', image)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

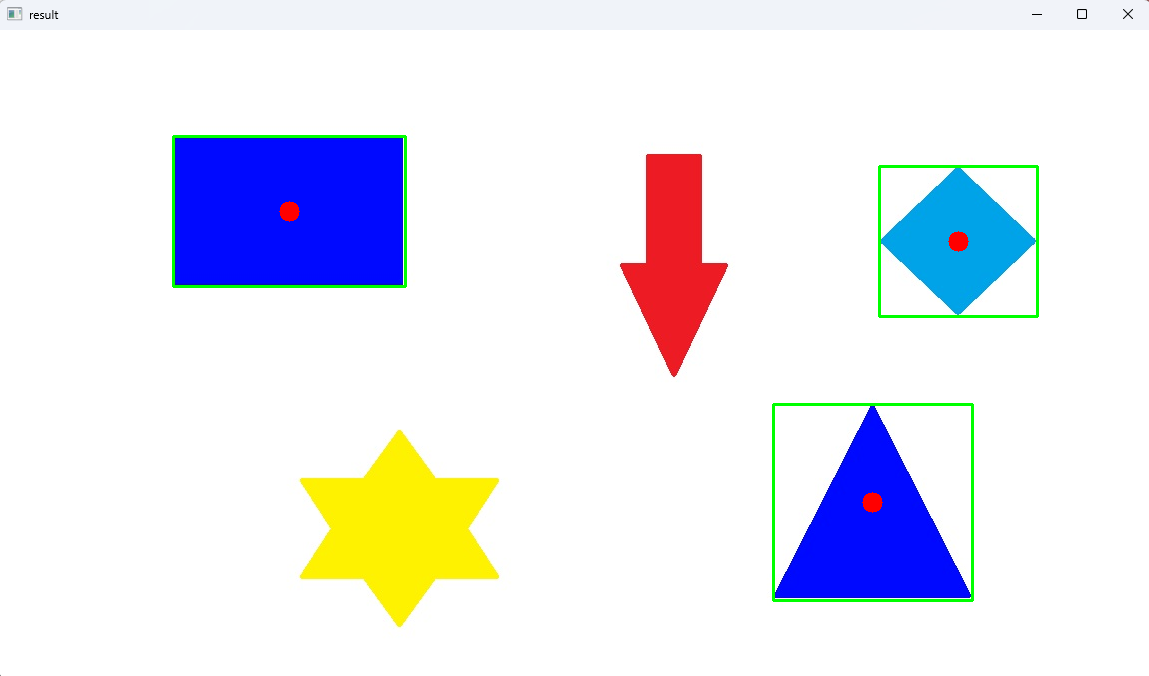


Рисунок 1 - Пример выполнения программы №1

**Заключение**

В процессе прохождения учебной практики были получены новые навыки сфере программирования, информатики и графики. Были решены различные логические и алгоритмические задачи, которые хорошо повлияли на восприятие новой теории. Также были задачи, связанные с виртуальным электронным устройством Arduino, где мы создавали схемы и приводили их в работу. Особое место имела задача, которая лежит в основе программного обеспечения видеокамер, интеллекта в машинах с автопилотом, в ней использовался алгоритм распознания объектов по заданным условиям. Умение общаться с такими знаниями может сыграть большую роль в будущем.

Также мы посетили две увлекательные экскурсии. Первая была проведена в филиале иркутского РДУ, месте, в котором происходит управление и надзор на электричеством области, где нам были рассказаны занимательные особенности электро-составляющей нашего региона и некоторые особенности в IT сфере, хоть и она не так важна в данной отрасли. Вторая же экскурсия проводилась в офисе IT-компании ISPsystem, которая занимается разработкой различного ПО и программ. Рассказ вели сами работники на понятном, доступном и простом языке так, что невозможно было заскучать. Мы узнали много интересных особенностей данной сферы, а также получили полезные советы, которые могут помочь как в учёбе, так и при поиске и устройстве на работу. В тот день была плохая погода, но поездка скрасила столь серый день.

В конечном итоге, практика оставила множество приятных воспоминаний, а также дала дополнительные знания, которые обязательно пригодятся в будущем, я уверен, что в будущем мы будем вспоминать её с теплом на душе.