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 Левченко Матвея Александровича, 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:

<https://www.superjob.ru/resume/programmist-razrabotchik-c-55732447.html>

Руководитель образовательной программы

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Изображение выглядит как шаблон, Графика, пиксель, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.Оценка по практике

(ФИО, подпись, дата)

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

Иркутск 2025

Ссылка на резюме на сайте hh.ru:

<https://irkutsk.hh.ru/resume/36d0d197ff0efd13800039ed1f593176734853>

**Индивидуальное задание на прохождение**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для | Левченко Матвея Александровича | | | | |
|  | (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 «\_28\_» июня 2025 г.

Цели и задачи прохождения практики:

Содержание практики, вопросы, подлежащие изучению:

Планируемые результаты практики:

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

**Задание №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.

Изображение выглядит как Шрифт, снимок экрана, число, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.

Приложение 1.

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

**Формат входных данных**

В первой строке дано целое число N - высота пирамиды, далее следуют N строк из чисел, разделённых пробелом (в каждой строке на 1 число больше, чем в предыдущей)

**Формат выходных данных**

Сумма чисел в пути с вершины до основания (одно число)

Последовательность участков маршрута (числа, разделённые пробелом)

**Решение:**

#include <iostream>

#include <random>

#include <vector>

#include <algorithm>

using namespace std;

vector<vector<int>> Rand(int n) {

vector<vector<int>> mountain;

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<> dist(1, 99);

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

vector<int> row;

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

row.push\_back(dist(rd));

}

mountain.push\_back(row);

}

for (const auto& row : mountain) {

for (const auto& num : row) {

cout << num << " ";

}

cout << endl;

}

return mountain;

}

void MinWay(vector<vector<int>> mountain, int n) {

if (n == 0) {

cout << "0" << endl;

return;

}

int res = mountain[0][0];;

vector<int> way;

way.push\_back(mountain[0][0]);

int current\_num = 0;

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

if (current\_num + 1 >= mountain[i].size()) {

res += mountain[i][current\_num];

way.push\_back(mountain[i][current\_num]);

}

else {

if (mountain[i][current\_num] > mountain[i][current\_num + 1] && mountain[i][current\_num + 1] != 0) {

res += mountain[i][current\_num + 1];

way.push\_back(mountain[i][current\_num + 1]);

current\_num++;

}

else {

res += mountain[i][current\_num];

way.push\_back(mountain[i][current\_num]);

}

}

}

cout << res << endl;

for (const auto& i : way) {

cout << i << " ";

}

cout << endl;

}

int main() {

int n = 0;

cin >> n;

MinWay(Rand(n), n);

}

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, черно-белый

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным.

Примеры входных/выходных данных для задания 1

**Задание № 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.

Изображение выглядит как круг, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.

Приложение 2.

**Формат входных данных**

В первой строке дано целое число N – количество узлов сети и M – число линий связи. Далее следуют M строк из чисел, разделённых пробелом (узлы, которые связывает данная линия)

**Формат выходных данных**

Число необходимых линий связи (одно число)

**Решение:**

#include <iostream>

#include <vector>

enum Color {

White,

Grey,

Black

};

std::vector<std::vector<int>> vertices;

std::vector<Color> colors;

void DFS(int index) {

colors[index] = Grey;

for (int ner : vertices[index]) {

if (colors[ner] == White) {

DFS(ner);

}

}

colors[index] = Black;

}

int countComponents(int n) {

int count = 0;

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

if (colors[i] == White) {

DFS(i);

count++;

}

}

return count;

}

int main() {

int n = 0;

int m = 0;

std::cin >> n >> m;

vertices.resize(n);

colors.resize(n, White);

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

int from = 0;

int to = 0;

std::cin >> from >> to;

from--;

to--;

vertices[from].push\_back(to);

vertices[to].push\_back(from);

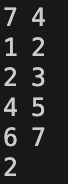
}

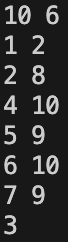
int components = countComponents(n);

std::cout << components - 1 << std::endl;

return 0;

}

 Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным. 

Примеры входных/выходных данных для задания 2

**Задание № 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.

**Формат входных данных**

Первая строка входных данных содержит два целых числа – n (1 или больше) и k (0 или больше) это количество чисел в массиве и количество запросов соответственно.

Следующие k строк содержат:

* либо 1 i x – Учетная информация о количестве, выпавшего на улице i (больше 0) x миллиметров снега.
* либо 2 u r – Запрос на подсчет количества снега на улицах от u до r (u и r больше 0 и могут быть равны друг другу)

**Формат выходных данных**

На каждый запрос второго типа надо вывести единственное число – суммарное выпавшего на них снега с момента начала наблюдения.

**Решение:**

#include <iostream>

#include <vector>

int main() {

int n = 0;

int k = 0;

std::cin >> n >> k;

std::vector<int> vec(n, 0);

std::vector<int> output;

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

int request = 0;

int a = 0;

int b = 0;

std::cin >> request >> a >> b;

if (request == 1) {

vec[a - 1] = b;

} else if (request == 2) {

int res = 0;

for (int j = a; j <= b; j++) {

res += vec[j - 1];

}

output.push\_back(res);

}

}

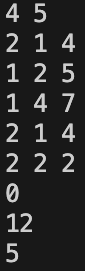
for (int res : output) {

std::cout << res << std::endl;

}

return 0;

}

 Изображение выглядит как текст, Шрифт, типография, число

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным.

Примеры входных/выходных данных для задания 3