

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)



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



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

Обучающегося _____
(ФИО, группа, подпись)

Руководитель практики от института ИТиАД

Кононенко Роман Владимирович, доцент
(ФИО, должность, подпись)

Руководитель образовательной
программы

Кононенко Р.В., доцент института ИТиАД
(ФИО, должность, подпись)

Оценка по практике _____
(ФИО, подпись, дата)

Содержание отчета на 4 стр.

Иркутск 2025

Индивидуальное задание на прохождение

учебной практики: технологической (проектно-технологической) практики

для Симонова Александра Андреевича

(ФИО обучающегося полностью)

обучающегося 1 курса

группы ИСИ6-24-1

по направлению подготовки Информационные системы и технологии
профиль Интеллектуальные системы обработки информации и управления

Место прохождения практики: ИРНТУ

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

« » 2023 г.

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

«16» июня 2025 г.

(подпись)

ДНЕВНИК

прохождения практики

обучающегося Симонова Александра Андреевича , 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
	16.06.2025	Выполнена задача №1, задача №2	
	17.06.2025	Выполнена задача №3	

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:

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <algorithm>
using namespace std;

vector<vector<int>> hill(int n) {
    vector<vector<int>> pyr(n);
    for (int i = 0; i < n; ++i) {
        pyr[i].resize(i + 1);
        for (int j = 0; j <= i; ++j) {
            pyr[i][j] = rand() % 100 + 1;
        }
    }
    return pyr;
}

pair<int, vector<int>> minPath(const vector<vector<int>>& pyr) {
    int n = pyr.size();
    vector<vector<int>> dp(n, vector<int>(n, 0));
    vector<vector<int>> par(n, vector<int>(n, -1));
    dp[0][0] = pyr[0][0];
    for (int i = 1; i < n; ++i) {
        for (int j = 0; j <= i; ++j) {
            if (j == 0) {
                dp[i][j] = dp[i - 1][j] + pyr[i][j];
                par[i][j] = j;
            } else if (j == i) {
                dp[i][j] = dp[i - 1][j - 1] + pyr[i][j];
                par[i][j] = j - 1;
            } else {
                if (dp[i - 1][j - 1] < dp[i - 1][j]) {
                    dp[i][j] = dp[i - 1][j - 1] + pyr[i][j];
                    par[i][j] = j - 1;
                } else {
                    dp[i][j] = dp[i - 1][j] + pyr[i][j];
                    par[i][j] = j;
                }
            }
        }
    }
}
```

```

    }
}
int minSum = dp[n - 1][0], minIdx = 0;
for (int j = 1; j < n; ++j) {
    if (dp[n - 1][j] < minSum) {
        minSum = dp[n - 1][j];
        minIdx = j;
    }
}
vector<int> path;
for (int i = n - 1; i >= 0; --i) {
    path.push_back(pyr[i][minIdx]);
    minIdx = par[i][minIdx];
}
reverse(path.begin(), path.end());
return {minSum, path};
}

int main() {
    cout << "Здравствуйте, введите высоту пирамиды: ";
    int n;
    cin >> n;
    rand();
    auto pyr = hill(n);

    for (const auto& row : pyr) {
        for (size_t i = 0; i < row.size(); ++i) {
            cout << row[i] << (i < row.size() - 1 ? " " : "");
        }
        cout << "\n";
    }
    auto result = minPath(pyr);
    cout << result.first << "\n";
    for (size_t i = 0; i < result.second.size(); ++i) {
        cout << result.second[i] << (i < result.second.size() - 1 ? " " : "");
    }
    cout << "\n";
    return 0;
}

```

Результат:

Здравствуйте, введите высоту пирамиды: 5

Гора:

87

78 16

94 36 87

93 50 22 63

28 91 60 64 27

Результат:

221

87 16 36 22 60

Здравствуйте, введите высоту пирамиды: 15

Гора:

87

78 16

94 36 87

93 50 22 63

28 91 60 64 27

41 27 73 37 12 69

68 30 83 31 63 24 68

36 30 3 23 59 70 68 94

57 12 43 30 74 22 20 85 38

99 25 16 71 14 27 92 81 57 74

63 71 97 82 6 26 85 28 37 6 47

30 14 58 25 96 83 46 15 68 35 65 44

51 88 9 77 79 89 85 4 52 55 100 33 61

77 69 40 13 27 87 95 40 96 71 35 79 68 2

98 3 18 93 53 57 2 81 87 42 66 90 45 20 41

Результат:

553

87 16 36 22 60 37 31 23 30 14 26 46 4 40 81

Здравствуйте, введите высоту пирамиды: 3

Гора:

87

78 16

94 36 87

Результат:

139

87 16 36

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:

```
#include <iostream>
#include <vector>
using namespace std;

vector<vector<int>> generateGraph(int n, int m) {
    vector<vector<int>> graph(n + 1);
    cout << "Введите " << m << " рёбер (формат: u v, где u и v <= "<< n << "):" << endl;
    for (int i = 0; i < m; ++i) {
        int u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }
    return graph;
}

void dfs(int v, vector<bool>& visited, const vector<vector<int>>& graph) {
    visited[v] = true;

    for (int neighbor : graph[v]) {
        if (!visited[neighbor]) {
            dfs(neighbor, visited, graph);
        }
    }
}

int count_communication_lines(int n, const vector<vector<int>>& graph) {
    vector<bool> visited(n+1, false);
    int count = -1;

    for (int i = 1; i < n+1; ++i) {
        if (!visited[i]) {
            dfs(i, visited, graph);
            ++count;
        }
    }
}
```



```

        return count;
    }

int main() {
    int n, m;
    cout << "Здравствуйте, введите число вершин графа (узлы сети, не более 1000)"
    << endl <<
    "и набор рёбер (сохранившиеся линии связи, не более 1000) через пробел и
    нажмите enter:";
    cin >> n;
    cin >> m;
    vector<vector<int>> graph = generateGraph(n,m);
    if (n <= 0 or m <= 0){
        cout << "неверные данные. Пожалуйста, введите количество вершин, превышающее
    0!" << endl;
        return 0;
    }
    int count = count_communication_lines(n, graph);
    cout << "минимальное число линий связи: "<<count << endl;
    return 0;
}

```

Результат:

```

Здравствуйте, введите число вершин графа (узлы сети, не более 1000)
и набор рёбер (сохранившиеся линии связи, не более 1000) через пробел и нажмите enter:10 6
Введите 6 рёбер (формат: u v, где u и v <= 10):
1 2
2 8
4 10
5 9
6 10
7 9
минимальное число линий связи: 3

```

```

Здравствуйте, введите число вершин графа (узлы сети, не более 1000)
и набор рёбер (сохранившиеся линии связи, не более 1000) через пробел и нажмите enter:4 3
Введите 3 рёбер (формат: u v, где u и v <= 4):
1 2
2 3
3 4
минимальное число линий связи: 0

```

```

Здравствуйте, введите число вершин графа (узлы сети, не более 1000)
и набор рёбер (сохранившиеся линии связи, не более 1000) через пробел и нажмите enter:10 6
Введите 6 рёбер (формат: u v, где u и v <= 10):
1 2
2 3
3 4
4 5
7 8
8 9
минимальное число линий связи: 3

```

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:

```
#include <iostream>
#include <vector>
using namespace std;

void snow_center(int n, int k){
    vector<int> snow_level(n,0);
    int count_k = 0;
    int command, a, b;
    while (count_k != k)
    {
        cin >> command >> a >> b;
        if (command == 1){
            snow_level[a-1] = b;
        } else if (command == 2){
            int snow_count = 0;
            for (int i = a-1; i <= b-1; i++){
                snow_count += snow_level[i];
            }
            cout << snow_count << endl;
        } else {
            cout << "Неправильные входные данные. они должны начинаться либо с 1, либо с 2." << endl;
            continue;
        }
    }
}
```

```

    }
    ++count_k;
}
}

int main() {
    int n,k;
    cout << "введите количество улиц и число запросов через пробел: ";
    cin >> n >> k;
    snow_center(n,k);
    return 0;
}

```

Результат:

```

введите количество улиц и число запросов через пробел: 3 8
1 3 1
1 1 3
2 1 3
1 2 7
2 2 3
1 3 9
1 3 7
2 1 3
4
8
27

```

```

введите количество улиц и число запросов через пробел: 6 5
2 1 6
1 3 2
2 2 4
1 6 3
2 1 6
0
2
5

```

```

введите количество улиц и число запросов через пробел: 5 3
1 3 7
1 1 4
2 1 5
11

```