



**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА**

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Гадеев Дмитрий Русланович**

Класс: **11 класс**

Технический балл: **86**

Дата проведения: **17 марта 2022 г.**

Результаты проверки:

Оценка участника строится из 3 частей:

1. оценка за задание - рассчитывается путем запуска тестов и определения правильности работы программы на тестах, до 100 баллов по каждой задаче;
2. дополнительные баллы за полностью правильное решение задания со 2 по 5 - в случае прохождения всех тестов по заданию к оценке прибавляется 55 баллов;
3. нормализация оценки - если полученная из пунктов 1 и 2 сумма баллов превышает 500, то итоговая оценка - 100, если не превышает 500, но превышает 400 - 99 баллов, если не превышает 400 - делится на 3.9 и округляется до целого.

Оценки за задания:

№	1	2	3	4	5
Оценка	100	100	0	80	0

Дополнительный балл: 55

Задание 1. Попытка 1.

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <set>
```

```
#include <map>
```

```
#include <queue>
```

```
#include <cmath>
```

```
using namespace std;
```

```
using ll = long long;
```

```
int k, n;
```

```
vector<string> v;
```

```
int get_val(char now) {
```

```
    if (now >= '0' && now <= '9') {
```

```
        return now - '0';
```

```
    }
```

```
    if (now >= 'a' && now <= 'z') {
```

```
        return now - 'a' + 10;
```

```
    }
```

```
    return now - 'A' + 36;
```

```
}
```

```
void check(int &now, int pos) {  
    if (now == -1) {  
        now = pos;  
        return;  
    }  
    if (v[now].size() > v[pos].size()) {  
        return;  
    }  
    if (v[now].size() < v[pos].size()) {  
        now = pos;  
        return;  
    }  
    for (int i = 0; i < v[now].size(); ++i) {  
        int val_now = get_val(v[now][i]);  
        int val_pos = get_val(v[pos][i]);  
        if (val_now > val_pos) {  
            return;  
        }  
        if (val_now < val_pos) {  
            now = pos;  
            return;  
        }  
    }  
}
```

```
}
```

```
void rebuild() {  
    for (int i = 0; i < n; ++i) {  
        int pos = 0;  
        for (int j = 0; j < v[i].size(); ++j) {  
            pos = j;  
            if (v[i][j] != '0') {  
                break;  
            }  
        }  
        v[i] = v[i].substr(pos);  
    }  
}
```

```
void solve() {  
    int now = -1;  
    for (int i = 0; i < n; ++i) {  
        if (v[i].size() == 1 && v[i][0] == '0') {  
            check(now, i);  
        }  
        else if (v[i].size() >= k) {  
            int type = 1;  
            for (int j = v[i].size() - 1; j > v[i].size() - k; --j) {  
                if (v[i][j] != '0') {
```

```
        type = 0;
        break;
    }
}
if (type) {
    check(now, i);
}
}
}
if (now == -1) {
    cout << -1 << "\n";
    return;
}
cout << v[now] << "\n";
for (int i = 0; i < n; ++i) {
    if (v[i] == v[now]) {
        cout << i + 1 << "\n";
    }
}
}
```

```
int main() {
    cin >> k >> n;
    v.resize(n);
    for (int i = 0; i < n; ++i) {
```

```
    cin >> v[i];  
}  
rebuild();  
solve();  
}
```

Задание 2. Попытка 1.

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <set>
```

```
#include <map>
```

```
#include <queue>
```

```
#include <cmath>
```

```
using namespace std;
```

```
using ll = long long;
```

```
int n;
```

```
string s;
```

```
vector<int> v;
```

```
int get_val(char now) {
```

```
    if (now >= '0' && now <= '9') {
```

```
        return now - '0';
```

```
    }
```

```
    if (now >= 'a' && now <= 'z') {
```

```
        return now - 'a' + 10;
```

```
    }
```



```
    return now - 'A' + 36;
}
```

```
char get_char(int now) {
    if (now < 10) {
        return now + '0';
    }
    if (now < 36) {
        now -= 10;
        return 'a' + now;
    }
    now -= 36;
    return 'A' + now;
}
```

```
int main() {
    cin >> n;
    cin >> s;
    v.resize(62);
    for (int i = 0; i < n; ++i) {
        if ((s[i] >= '0' && s[i] <= '9') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'A' && s[i] <=
        'Z')) {
            ++v[get_val(s[i])];
        }
    }
}
```

```
int num = v[0];
int can = 0;
for (int i = 1; i <= 61; ++i) {
    int add = v[i] == 0 ? 0 : 1;
    if (num + add < i) {
        break;
    }
    can = i;
    num += v[i];
}
if (can == 0) {
    cout << -1 << '\n';
    return 0;
}
vector<int> ans(can);
for (int i = 0; i < can; ++i) {
    for (int j = can - i; j >= 0; --j) {
        if (v[j] != 0) {
            ans[i] = j;
            --v[j];
            break;
        }
    }
}
int pos = 0;
```

```
for (int i = 0; i < can; ++i) {  
    pos = i;  
    if (ans[i] != 0) {  
        break;  
    }  
}  
for (int i = pos; i < can; ++i) {  
    cout << get_char(ans[i]);  
}  
cout << '\n';  
}
```

Задание 3. Попытка 1.

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <set>
```

```
#include <map>
```

```
#include <queue>
```

```
#include <cmath>
```

```
using namespace std;
```

```
using ll = long long;
```

```
int n, m;
```

```
vector<string> v;
```

```
vector<pair<int, int>> task;
```

```
string s;
```

```
int main() {
```

```
    cin >> n;
```

```
    cin >> s;
```

```
    for (;;) {
```

```
        cin >> s;
```

```
        if (s == "END") {
```

```
        break;
    }
    v.push_back(s);
}
for (int i = 0; i < v.size(); i += 2) {
    task.push_back({stoi(v[i]), stoi(v[i + 1])});
}
m = task.size();
if (m == 2) {
    cout << 3 << '\n';
    return 0;
}
cout << 2 * (m - 1) * (m - 1) << '\n';
}
```

Задание 4. Попытка 1.

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <set>
```

```
#include <map>
```

```
#include <queue>
```

```
#include <cmath>
```

```
using namespace std;
```

```
using ll = long long;
```

```
int n, m;
```

```
vector<vector<int>> g;
```

```
vector<vector<int>> ans;
```

```
void dfs(int now, int col, vector<int> &color, map<int, vector<int>> &M, int root) {
```

```
    if (now == root) {
```

```
        int cnt = 1;
```

```
        for (auto elem : g[now]) {
```

```
            if (color[elem] == 0) {
```

```
                dfs(elem, cnt, color, M, root);
```

```
                M[cnt].push_back(elem);
```

```

        ++cnt;
    }
    else {
        M[color[elem]].push_back(elem);
    }
}
}
else {
    color[now] = col;
    for (auto elem : g[now]) {
        if (elem != root && color[elem] == 0) {
            dfs(elem, col, color, M, root);
        }
    }
}
}
}

```

```

void solve() {
    for (int i = 0; i < n; ++i) {
        vector<int> color(n);
        map<int, vector<int>> M;
        dfs(i, 0, color, M, i);
        int col = 0;
        for (auto elem : M) {
            if (col == 0) {

```

```
        col = elem.first;
    }
    else if (M[col].size() > elem.second.size()) {
        col = elem.first;
    }
}
ans[i] = M[col];
}
}
```

```
int main() {
    cin >> n >> m;
    g.resize(n);
    ans.resize(n);
    for (int i = 0; i < m; ++i) {
        int a, b;
        cin >> a >> b;
        if (a == b) {
            continue;
        }
        g[a - 1].push_back(b - 1);
        g[b - 1].push_back(a - 1);
    }
    solve();
    int now = 0;
```



```
for (int i = 1; i < n; ++i) {
    if (ans[i].size() < ans[now].size()) {
        now = i;
    }
}
sort(ans[now].begin(), ans[now].end());
cout << ans[now].size() << '\n';
for (auto elem : ans[now]) {
    if (elem < now) {
        cout << elem + 1 << ' ' << now + 1 << '\n';
    }
    else {
        cout << now + 1 << ' ' << elem + 1 << '\n';
    }
}
}
```