



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

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Асташкин Фёдор Кириллович**

Класс: **11 класс**

Технический балл: **71**

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	96	0	80	2

Дополнительный балл: 0

Задание 1. Попытка 1.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
inline int get(char ch) {  
    if ('0' <= ch && ch <= '9') {  
        return ch - '0';  
    }  
    else if ('a' <= ch && ch <= 'z') {  
        return 10 + (ch - 'a');  
    }  
    else {  
        return 36 + (ch - 'A');  
    }  
}
```

```
inline vector<int> to_normal(const string& s) {  
    vector<int> a;  
    int it = 0;  
    while (s[it] == '0' && it < s.size() - 1) {  
        ++it;  
    }  
    for (int i = it; i < s.size(); ++i) {  
        a.push_back(get(s[i]));  
    }  
    return a;  
}
```

```
struct FBC {  
    string sl;  
    vector<int> a;  
    int n = 0;  
  
    FBC() {}  
  
    FBC(const string& s) {  
        sl = s;  
        a = to_normal(s);  
        n = a.size();  
    }  
};
```

```
inline bool operator > (const FBC& a, const FBC& b) {  
    if (a.n > b.n) {  
        return 1;  
    }  
    else if (a.n < b.n) {  
        return 0;  
    }  
    else {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] > b.a[i]) {  
                return 1;  
            }  
        }  
        return 0;  
    }  
}
```

```
inline bool operator == (const FBC& a, const FBC& b) {  
    if (a.n == b.n) {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] != b.a[i]) {  
                return 0;  
            }  
        }  
    }  
}
```

```
        return 1;
    }
    else {
        return 0;
    }
}
```

```
inline char get_back(int x) {
    if (0 <= x && x <= 9) {
        return '0' + x;
    }
    else if (10 <= x && x <= 35) {
        return 'a' + (x - 10);
    }
    else {
        return 'A' + (x - 36);
    }
}
```

```
inline string to_back(const FBC& a) {
    string s(a.n, '0');
    for (int i = 0; i < a.n; ++i) {
        s[i] = get_back(a.a[i]);
    }
    return s;
}
```

```
}
```

```
inline bool is_good(const FBC& a, int k) {  
    if (a.n == 1 && a.a[0] == 0) {  
        return 1;  
    }  
    if (k > a.n) {  
        return 0;  
    }  
    for (int i = a.n - k + 1; i < a.n; ++i) {  
        if (a.a[i] != 0) {  
            return 0;  
        }  
    }  
    return 1;  
}
```

```
signed main() {  
    ll k, n;  
    cin >> k >> n;  
    vector<FBC> a(n);  
    FBC ma = FBC("0");  
    bool flag = 0;  
    for (int i = 0; i < n; ++i) {  
        string s;
```

```
    cin >> s;
    a[i] = FBC(s);
    if (is_good(a[i], k)) {
        if (flag == 0) {
            flag = 1;
            ma = a[i];
        }
        else if (a[i] > ma) {
            ma = a[i];
        }
    }
}
if (flag == 0) {
    cout << -1;
}
else {
    cout << to_back(ma) << '\n';
    for (int i = 0; i < n; ++i) {
        if (a[i] == ma) {
            cout << i + 1 << '\n';
        }
    }
}
return 0;
}
```


Задание 1. Попытка 2.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
inline int get(char ch) {  
    if ('0' <= ch && ch <= '9') {  
        return ch - '0';  
    }  
    else if ('a' <= ch && ch <= 'z') {  
        return 10 + (ch - 'a');  
    }  
    else {  
        return 36 + (ch - 'A');  
    }  
}
```

```
inline vector<int> to_normal(const string& s) {  
    vector<int> a;  
    int it = 0;  
    while (s[it] == '0' && it < s.size() - 1) {  
        ++it;  
    }  
    for (int i = it; i < s.size(); ++i) {  
        a.push_back(get(s[i]));  
    }  
    return a;  
}
```

```
struct FBC {  
    string sl;  
    vector<int> a;  
    int n = 0;  
  
    FBC() {}  
  
    FBC(const string& s) {  
        sl = s;  
        a = to_normal(s);  
        n = a.size();  
    }  
};
```

```
inline bool operator > (const FBC& a, const FBC& b) {  
    if (a.n > b.n) {  
        return 1;  
    }  
    else if (a.n < b.n) {  
        return 0;  
    }  
    else {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] > b.a[i]) {  
                return 1;  
            }  
            else if (a.a[i] < b.a[i]) {  
                return 0;  
            }  
        }  
        return 0;  
    }  
}
```

```
inline bool operator == (const FBC& a, const FBC& b) {  
    if (a.n == b.n) {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] != b.a[i]) {
```

```
        return 0;
    }
}
return 1;
}
else {
    return 0;
}
}
```

```
inline char get_back(int x) {
    if (0 <= x && x <= 9) {
        return '0' + x;
    }
    else if (10 <= x && x <= 35) {
        return 'a' + (x - 10);
    }
    else {
        return 'A' + (x - 36);
    }
}
```

```
inline string to_back(const FBC& a) {
    string s(a.n, '0');
    for (int i = 0; i < a.n; ++i) {
```

```
        s[i] = get_back(a.a[i]);
    }
    return s;
}
```

```
inline bool is_good(const FBC& a, int k) {
    if (a.n == 1 && a.a[0] == 0) {
        return 1;
    }
    if (k > a.n) {
        return 0;
    }
    for (int i = a.n - k + 1; i < a.n; ++i) {
        if (a.a[i] != 0) {
            return 0;
        }
    }
    return 1;
}
```

```
signed main() {
    ll k, n;
    cin >> k >> n;
    vector<FBC> a(n);
    FBC ma = FBC("0");
```

```

bool flag = 0;
for (int i = 0; i < n; ++i) {
    string s;
    cin >> s;
    a[i] = FBC(s);
    if (is_good(a[i], k)) {
        if (flag == 0) {
            flag = 1;
            ma = a[i];
        }
        else if (a[i] > ma) {
            ma = a[i];
        }
    }
}
if (flag == 0) {
    cout << -1;
}
else {
    cout << to_back(ma) << '\n';
    for (int i = 0; i < n; ++i) {
        if (a[i] == ma) {
            cout << i + 1 << '\n';
        }
    }
}

```

```
    }  
    return 0;  
}
```


Задание 1. Попытка 3.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
inline int get(char ch) {  
    if ('0' <= ch && ch <= '9') {  
        return ch - '0';  
    }  
    else if ('a' <= ch && ch <= 'z') {  
        return 10 + (ch - 'a');  
    }  
    else {  
        return 36 + (ch - 'A');  
    }  
}
```

```
inline vector<int> to_normal(const string& s) {  
    vector<int> a;  
    int it = 0;  
    while (s[it] == '0' && it < s.size() - 1) {  
        ++it;  
    }  
    for (int i = it; i < s.size(); ++i) {  
        a.push_back(get(s[i]));  
    }  
    return a;  
}
```

```
struct FBC {  
    //string sl;  
    vector<int> a;  
    int n = 0;  
  
    FBC() {}  
  
    FBC(const string& s) {  
        //sl = s;  
        a = to_normal(s);  
        n = a.size();  
    }  
};
```

```
inline bool operator > (const FBC& a, const FBC& b) {  
    if (a.n > b.n) {  
        return 1;  
    }  
    else if (a.n < b.n) {  
        return 0;  
    }  
    else {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] > b.a[i]) {  
                return 1;  
            }  
            else if (a.a[i] < b.a[i]) {  
                return 0;  
            }  
        }  
        return 0;  
    }  
}
```

```
inline bool operator == (const FBC& a, const FBC& b) {  
    if (a.n == b.n) {  
        for (int i = 0; i < a.n; ++i) {  
            if (a.a[i] != b.a[i]) {
```

```
        return 0;
    }
}
return 1;
}
else {
    return 0;
}
}
```

```
inline char get_back(int x) {
    if (0 <= x && x <= 9) {
        return '0' + x;
    }
    else if (10 <= x && x <= 35) {
        return 'a' + (x - 10);
    }
    else {
        return 'A' + (x - 36);
    }
}
```

```
inline string to_back(const FBC& a) {
    string s(a.n, '0');
    for (int i = 0; i < a.n; ++i) {
```

```
        s[i] = get_back(a.a[i]);
    }
    return s;
}
```

```
inline bool is_good(const FBC& a, int k) {
    if (a.n == 1 && a.a[0] == 0) {
        return 1;
    }
    if (k > a.n) {
        return 0;
    }
    for (int i = a.n - k + 1; i < a.n; ++i) {
        if (a.a[i] != 0) {
            return 0;
        }
    }
    return 1;
}
```

```
signed main() {
    ll k, n;
    cin >> k >> n;
    vector<FBC> a(n);
    FBC ma = FBC("0");
```

```

bool flag = 0;
for (int i = 0; i < n; ++i) {
    string s;
    cin >> s;
    a[i] = FBC(s);
    if (is_good(a[i], k)) {
        if (flag == 0) {
            flag = 1;
            ma = a[i];
        }
        else if (a[i] > ma) {
            ma = a[i];
        }
    }
}
if (flag == 0) {
    cout << -1;
}
else {
    cout << to_back(ma) << '\n';
    for (int i = 0; i < n; ++i) {
        if (a[i] == ma) {
            cout << i + 1 << '\n';
        }
    }
}

```

```
    }  
    return 0;  
}
```

Задание 2. Попытка 1.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <deque>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
inline int get(char ch) {  
    if ('0' <= ch && ch <= '9') {  
        return ch - '0';  
    }  
    else if ('a' <= ch && ch <= 'z') {  
        return 10 + (ch - 'a');  
    }  
    else {  
        return 36 + (ch - 'A');  
    }  
}
```



```
inline vector<int> to_normal(const string& s) {  
    vector<int> a;  
    int it = 0;  
    while (s[it] == '0' && it < s.size() - 1) {  
        ++it;  
    }  
    for (int i = it; i < s.size(); ++i) {  
        a.push_back(get(s[i]));  
    }  
    return a;  
}
```

```
struct FBC {  
    //string sl;  
    vector<int> a;  
    int n = 0;  
  
    FBC() {}  
  
    FBC(const string& s) {  
        //sl = s;  
        a = to_normal(s);  
        n = a.size();  
    }  
}
```

```
};
```

```
inline bool operator > (const FBC& a, const FBC& b) {
```

```
    if (a.n > b.n) {
```

```
        return 1;
```

```
    }
```

```
    else if (a.n < b.n) {
```

```
        return 0;
```

```
    }
```

```
    else {
```

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

```
            if (a.a[i] > b.a[i]) {
```

```
                return 1;
```

```
            }
```

```
            else if (a.a[i] < b.a[i]) {
```

```
                return 0;
```

```
            }
```

```
        }
```

```
        return 0;
```

```
    }
```

```
}
```

```
inline bool operator == (const FBC& a, const FBC& b) {
```

```
    if (a.n == b.n) {
```

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

```
        if (a.a[i] != b.a[i]) {
            return 0;
        }
    }
    return 1;
}
else {
    return 0;
}
}
```

```
inline char get_back(int x) {
    if (0 <= x && x <= 9) {
        return '0' + x;
    }
    else if (10 <= x && x <= 35) {
        return 'a' + (x - 10);
    }
    else {
        return 'A' + (x - 36);
    }
}
```

```
inline string to_back(const FBC& a) {
    string s(a.n, '0');
```

```
    for (int i = 0; i < a.n; ++i) {  
        s[i] = get_back(a.a[i]);  
    }  
    return s;  
}
```

```
inline bool is_good(const FBC& a, int k) {  
    if (a.n == 1 && a.a[0] == 0) {  
        return 1;  
    }  
    if (k > a.n) {  
        return 0;  
    }  
    for (int i = a.n - k + 1; i < a.n; ++i) {  
        if (a.a[i] != 0) {  
            return 0;  
        }  
    }  
    return 1;  
}
```

```
inline bool is_symb(char ch) {  
    return ('0' <= ch && ch <= '9') || ('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z');  
}
```

```
signed main() {  
    ll n;  
    cin >> n;  
    string s;  
    cin >> s;  
    string t;  
    for (char ch : s) {  
        if (is_symb(ch)) {  
            t.push_back(ch);  
        }  
    }  
    if (t.size() == 0) {  
        cout << -1;  
    }  
    else {  
        sort(t.rbegin(), t.rend());  
        const int SZ = 61;  
        FBC a(t);  
        sort(a.a.begin(), a.a.end());  
        if (a.a[0] > 1) {  
            cout << -1;  
            return 0;  
        }  
        vector<int> d;
```

```
for (int i = 0; i < a.n && i < SZ; ++i) {
    if (a.a[i] <= i + 1) {
        d.push_back(a.a[i]);
    }
}
int it = a.n - 1;
for (int i = d.size() - 1; i >= 0 && it >= 0; --i) {
    while (it >= 0 && a.a[it] > i + 1) {
        --it;
    }
    if (it >= 0) {
        d[i] = a.a[it];
        --it;
    }
}
reverse(d.begin(), d.end());
a.a = d;
a.n = d.size();
cout << to_back(a);
}
return 0;
}
```

Задание 2. Попытка 2.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
#include <deque>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
inline int get(char ch) {  
    if ('0' <= ch && ch <= '9') {  
        return ch - '0';  
    }  
    else if ('a' <= ch && ch <= 'z') {  
        return 10 + (ch - 'a');  
    }  
    else {  
        return 36 + (ch - 'A');  
    }  
}
```

```
inline vector<int> to_normal(const string& s) {  
    vector<int> a;  
    int it = 0;  
    while (s[it] == '0' && it < s.size() - 1) {  
        ++it;  
    }  
    for (int i = it; i < s.size(); ++i) {  
        a.push_back(get(s[i]));  
    }  
    return a;  
}
```

```
struct FBC {  
    //string sl;  
    vector<int> a;  
    int n = 0;  
  
    FBC() {}  
  
    FBC(const string& s) {  
        //sl = s;  
        a = to_normal(s);  
        n = a.size();  
    }  
}
```



```
};
```

```
inline bool operator > (const FBC& a, const FBC& b) {
```

```
    if (a.n > b.n) {
```

```
        return 1;
```

```
    }
```

```
    else if (a.n < b.n) {
```

```
        return 0;
```

```
    }
```

```
    else {
```

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

```
            if (a.a[i] > b.a[i]) {
```

```
                return 1;
```

```
            }
```

```
            else if (a.a[i] < b.a[i]) {
```

```
                return 0;
```

```
            }
```

```
        }
```

```
        return 0;
```

```
    }
```

```
}
```

```
inline bool operator == (const FBC& a, const FBC& b) {
```

```
    if (a.n == b.n) {
```

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

```
        if (a.a[i] != b.a[i]) {
            return 0;
        }
    }
    return 1;
}
else {
    return 0;
}
}
```

```
inline char get_back(int x) {
    if (0 <= x && x <= 9) {
        return '0' + x;
    }
    else if (10 <= x && x <= 35) {
        return 'a' + (x - 10);
    }
    else {
        return 'A' + (x - 36);
    }
}
```

```
inline string to_back(const FBC& a) {
    string s(a.n, '0');
```

```
    for (int i = 0; i < a.n; ++i) {  
        s[i] = get_back(a.a[i]);  
    }  
    return s;  
}
```

```
inline bool is_good(const FBC& a, int k) {  
    if (a.n == 1 && a.a[0] == 0) {  
        return 1;  
    }  
    if (k > a.n) {  
        return 0;  
    }  
    for (int i = a.n - k + 1; i < a.n; ++i) {  
        if (a.a[i] != 0) {  
            return 0;  
        }  
    }  
    return 1;  
}
```

```
inline bool is_symb(char ch) {  
    return ('0' <= ch && ch <= '9') || ('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z');  
}
```

```
signed main() {  
    ll n;  
    cin >> n;  
    string s;  
    getchar();  
    getline(cin, s);  
    string t;  
    for (char ch : s) {  
        if (is_symb(ch)) {  
            t.push_back(ch);  
        }  
    }  
    if (t.size() == 0) {  
        cout << -1;  
    }  
    else {  
        sort(t.rbegin(), t.rend());  
        const int SZ = 61;  
        FBC a(t);  
        sort(a.a.begin(), a.a.end());  
        if (a.a[0] > 1) {  
            cout << -1;  
            return 0;  
        }  
    }  
}
```

```
vector<int> d;

for (int i = 0; i < a.n && i < SZ; ++i) {
    if (a.a[i] <= i + 1) {
        d.push_back(a.a[i]);
    }
}

int it = a.n - 1;

for (int i = d.size() - 1; i >= 0 && it >= 0; --i) {
    while (it >= 0 && a.a[it] > i + 1) {
        --it;
    }
    if (it >= 0) {
        d[i] = a.a[it];
        --it;
    }
}

reverse(d.begin(), d.end());

a.a = d;

a.n = d.size();

cout << to_back(a);

}

return 0;

}
```

Задание 4. Попытка 1.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
vector<vector<int>> gr;
```

```
int n, m;
```

```
inline int count(int mask) {
```

```
    int ans = 0;
```

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

```
        if ((mask >> i) & 1) {
```

```
            for (int u : gr[i]) {
```

```
                if (((mask >> u) & 1) == 0) {
```

```
                    ++ans;
```

```
                }
```

```
            }
```

```

        }
    }
    return ans;
}

```

```

inline void getans(int mask) {
    vector<pair<int, int>> ans;
    for (int i = 0; i < n; ++i) {
        if ((mask >> i) & 1) {
            for (int u : gr[i]) {
                if (((mask >> u) & 1) == 0) {
                    ans.push_back({ min(i, u), max(i, u) });
                }
            }
        }
    }
    sort(ans.begin(), ans.end());
    for (auto i : ans) {
        cout << i.first + 1 << ' ' << i.second + 1 << '\n';
    }
}

```

```

signed main() {
    cin >> n >> m;
    gr.resize(n);

```

```

for (int i = 0; i < m; ++i) {
    int v, u;
    cin >> v >> u;
    if (v != u) {
        --v; --u;
        gr[v].push_back(u);
        gr[u].push_back(v);
    }
}
if (n == 1) {
    cout << 0;
    return 0;
}
int maskans = 0, ans = m + 1;
for (int mask = 1; mask < (1 << n) - 1; ++mask) {
    int temp = count(mask);
    if (temp < ans) {
        ans = temp;
        maskans = mask;
    }
}
cout << ans << '\n';
getans(maskans);
return 0;
}

```


Задание 4. Попытка 2.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

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

```
vector<vector<int>> gr, num;
```

```
vector<bool>used;
```

```
vector<int> h, up/*, par*/;
```

```
int n, m;
```

```
inline int count(int mask) {
```

```
    int ans = 0;
```

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

```
        if ((mask >> i) & 1) {
```

```
            for (int u : gr[i]) {
```

```
                if (((mask >> u) & 1) == 0) {
```

```

        ++ans;
    }
}
}
}
return ans;
}

```

```

inline void getans(int mask) {
    vector<pair<int, int>> ans;
    for (int i = 0; i < n; ++i) {
        if ((mask >> i) & 1) {
            for (int u : gr[i]) {
                if (((mask >> u) & 1) == 0) {
                    ans.push_back({ min(i, u), max(i, u) });
                }
            }
        }
    }
    sort(ans.begin(), ans.end());
    for (auto i : ans) {
        cout << i.first + 1 << ' ' << i.second + 1 << "\n";
    }
}

```

```

int v1 = -1, u1 = -1;

map<int, map<int, int>> mapa;

inline void dfs(int v, int parent) {
    used[v] = 1;
    if (mapa[v][parent] > 1) {
        up[v] = h[parent];
    }
    for (int u : gr[v]) {
        if (u == parent) {
            continue;
        }
        else if (used[u]) {
            up[v] = min(up[v], h[u]);
        }
        else {
            up[u] = h[u] = h[v] + 1;
            dfs(u, v);
            up[v] = min(up[v], up[u]);
        }
    }
    if (v != 0 && up[v] == h[v]) {
        v1 = v;
        u1 = parent;
    }
}

```

```
}
```

```
signed main() {  
    cin >> n >> m;  
    gr.resize(n);  
    h.resize(n, 0);  
    used.resize(n, 0);  
    //par.resize(n);  
    num.resize(n);  
    up.resize(n, 0);  
    e.resize(m);  
    for (int i = 0; i < m; ++i) {  
        int v, u;  
        cin >> v >> u;  
        if (v != u) {  
            --v; --u;  
            e[i] = { v, u };  
            gr[v].push_back(u);  
            gr[u].push_back(v);  
            ++mapa[v][u];  
            ++mapa[u][v];  
            num[v].push_back(i);  
            num[u].push_back(i);  
        }  
    }  
}
```

```

if (n == 1) {
    cout << 0;
    return 0;
}
for (int i = 0; i < n; ++i) {
    if (gr[i].size() == 1) {
        cout << 1 << '\n';
        cout << min(i, gr[i][0]) + 1 << ' ' << max(i, gr[i][0]) + 1;
        return 0;
    }
}
//МОСТ
v1 = -1, u1 = -1;
dfs(0, 0);
if (v1 != -1) {
    cout << 1 << '\n';
    cout << min(v1, u1) + 1 << ' ' << max(v1, u1) + 1;
    return 0;
}
for (int i = 0; i < n; ++i) {
    if (gr[i].size() == 2) {
        cout << 2 << '\n';
        vector<pair<int, int>> ans = { e[num[i][0]], e[num[i][1]] };
        for (auto& i : ans) {
            if (i.first > i.second) {

```

```

        swap(i.first, i.second);
    }
}
sort(ans.begin(), ans.end());
for (auto& i : ans) {
    cout << i.first + 1 << ' ' << i.second + 1 << '\n';
}
return 0;
}
}
int maskans = 0, ans = m + 1;
for (int mask = 1; mask < (1 << n) - 1; ++mask) {
    int temp = count(mask);
    if (temp < ans) {
        ans = temp;
        maskans = mask;
    }
}
cout << ans << '\n';
getans(maskans);

```

//если есть мост или верш со степ1 то ответ 1

//или если есть верш со ст 2, то ответ 2

```
return 0;
```

```
}
```


Задание 4. Попытка 3.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

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

```
vector<vector<int>> gr, num;
```

```
vector<bool>used;
```

```
vector<int> h, up/*, par*/;
```

```
int n, m;
```

```
inline int count(int mask) {
```

```
    int ans = 0;
```

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

```
        if ((mask >> i) & 1) {
```

```
            for (int u : gr[i]) {
```

```
                if (((mask >> u) & 1) == 0) {
```

```

        ++ans;
    }
}
}
}
return ans;
}

```

```

inline void getans(int mask) {
    vector<pair<int, int>> ans;
    for (int i = 0; i < n; ++i) {
        if ((mask >> i) & 1) {
            for (int u : gr[i]) {
                if (((mask >> u) & 1) == 0) {
                    ans.push_back({ min(i, u), max(i, u) });
                }
            }
        }
    }
    sort(ans.begin(), ans.end());
    for (auto i : ans) {
        cout << i.first + 1 << ' ' << i.second + 1 << "\n";
    }
}

```

```

int v1 = -1, u1 = -1;

map<int, map<int, int>> mapa;

inline void dfs(int v, int parent) {
    used[v] = 1;
    if (mapa[v][parent] > 1) {
        up[v] = h[parent];
    }
    for (int u : gr[v]) {
        if (u == parent) {
            continue;
        }
        else if (used[u]) {
            up[v] = min(up[v], h[u]);
        }
        else {
            up[u] = h[u] = h[v] + 1;
            dfs(u, v);
            up[v] = min(up[v], up[u]);
        }
    }
    if (v != 0 && up[v] == h[v]) {
        v1 = v;
        u1 = parent;
    }
}

```

```
}
```

```
signed main() {  
    cin >> n >> m;  
    gr.resize(n);  
    h.resize(n, 0);  
    used.resize(n, 0);  
    //par.resize(n);  
    num.resize(n);  
    up.resize(n, 0);  
    e.resize(m);  
    for (int i = 0; i < m; ++i) {  
        int v, u;  
        cin >> v >> u;  
        if (v != u) {  
            --v; --u;  
            e[i] = { v, u };  
            gr[v].push_back(u);  
            gr[u].push_back(v);  
            ++mapa[v][u];  
            ++mapa[u][v];  
            num[v].push_back(i);  
            num[u].push_back(i);  
        }  
    }  
}
```

```

if (n == 1) {
    cout << 0;
    return 0;
}
for (int i = 0; i < n; ++i) {
    if (gr[i].size() == 1) {
        cout << 1 << '\n';
        cout << min(i, gr[i][0]) + 1 << ' ' << max(i, gr[i][0]) + 1;
        return 0;
    }
}
//МОСТ
v1 = -1, u1 = -1;
dfs(0, 0);
if (v1 != -1) {
    cout << 1 << '\n';
    cout << min(v1, u1) + 1 << ' ' << max(v1, u1) + 1;
    return 0;
}
for (int i = 0; i < n; ++i) {
    if (gr[i].size() == 2) {
        cout << 2 << '\n';
        vector<pair<int, int>> ans = { e[num[i][0]], e[num[i][1]] };
        for (auto& i : ans) {
            if (i.first > i.second) {

```

```

        swap(i.first, i.second);
    }
}
sort(ans.begin(), ans.end());
for (auto& i : ans) {
    cout << i.first + 1 << ' ' << i.second + 1 << "\n";
}
return 0;
}
}
if (n <= 20) {
    int maskans = 0, ans = m + 1;
    for (int mask = 1; mask < (1 << n) - 1; ++mask) {
        int temp = count(mask);
        if (temp < ans) {
            ans = temp;
            maskans = mask;
        }
    }
    cout << ans << "\n";
    getans(maskans);
}
else {
    int it = 0;
    for (int i = 0; i < n; ++i) {

```

```

        if (gr[i].size() < gr[it].size()) {
            it = i;
        }
    }
    cout << gr[it].size() << "\n";
    vector<pair<int, int>> ans(gr[it].size());
    for (int i = 0; i < gr[it].size(); ++i) {
        ans[i] = e[num[it][i]];
    }
    for (auto& i : ans) {
        if (i.first > i.second) {
            swap(i.first, i.second);
        }
    }
    sort(ans.begin(), ans.end());
    for (auto& i : ans) {
        cout << i.first + 1 << ' ' << i.second + 1 << "\n";
    }
}

//если есть мост или верш со степ1 то ответ 1
//или если есть верш со ст 2, то ответ 2

return 0;
}

```


Задание 5. Попытка 1.

```
#include <iostream>
```

```
#include <set>
```

```
#include <map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <string>
```

```
using namespace std;
```

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

```
using ld = long double;
```

```
signed main() {
```

```
    ll a1, b1, a2, b2;
```

```
    cin >> a1 >> b1 >> a2 >> b2;
```

```
    if (a1 == a2) {
```

```
        cout << abs(b1 - b2) + 1;
```

```
    }
```

```
    else if (a1 == 0 && b1 == 2 && a2 == 1 && b2 == 1) {
```

```
        cout << 2;
```

```
    }
```

```
    else {
```

```
        ll ans = abs(b1 - b2) + 1;
```

```
        ans += abs(a1 - a2) * 2;
```

```
        --ans;  
        cout << ans;  
    }  
    return 0;  
}
```

Председателю апелляционной комиссии
олимпиады школьников «Ломоносов»
Ректору МГУ имени М.В. Ломоносова
академику В.А. Садовничему

11"В" класс
ГБОУ школа №1568, пр. Шокальского д.7 к.2
Асташкин Фёдор Кириллович

апелляция.

Прошу пересмотреть выставленные технические баллы 71 за мою работу заключительного этапа по информатике.

31.03.2022